Digital Image Processing & Machine Vision

Lecture # 10

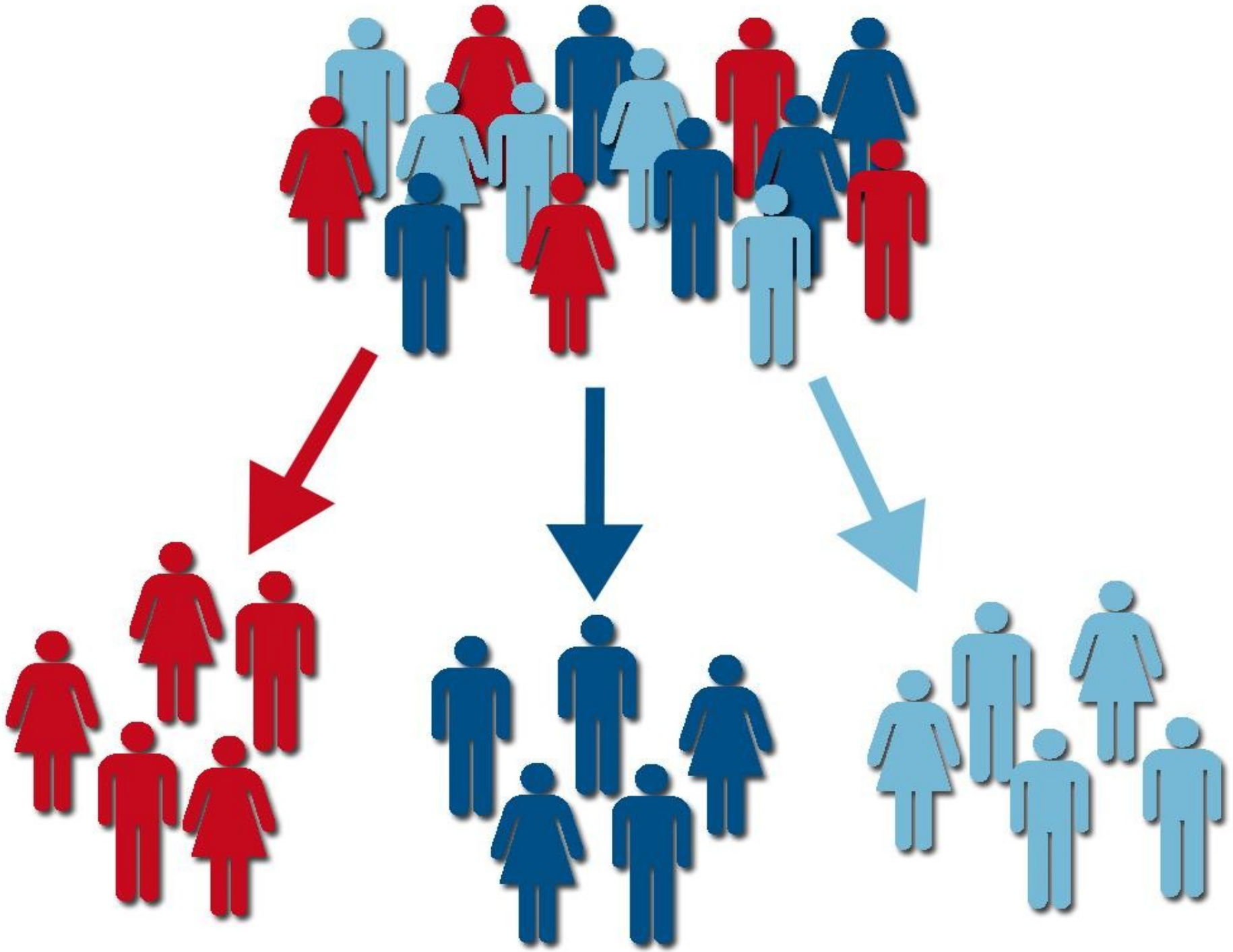
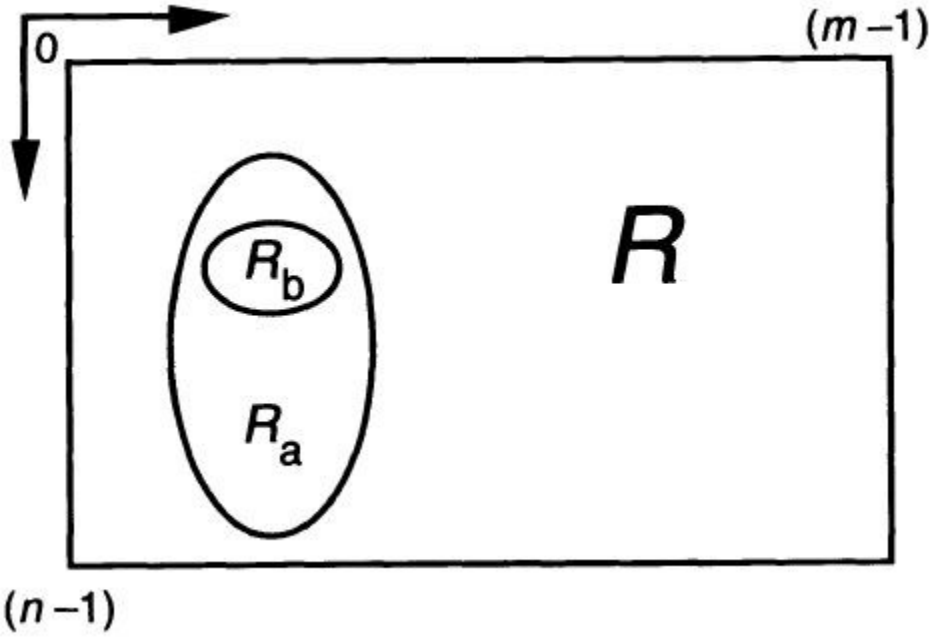


Image Segmenta


0
(m –1)
R
R_b
R_a
(n –1)

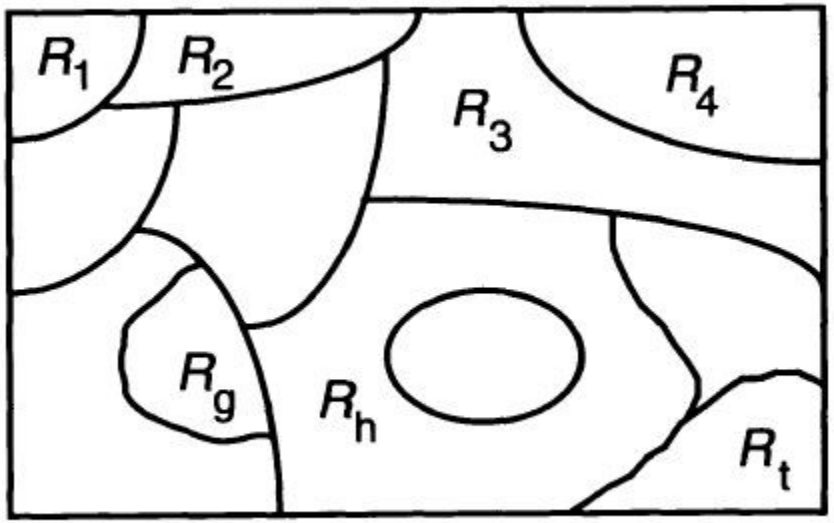

R_1
R_2
R_3
R_4
R_g
R_h
R_t

Image Segmentation

- To partition an image into meaningful regions

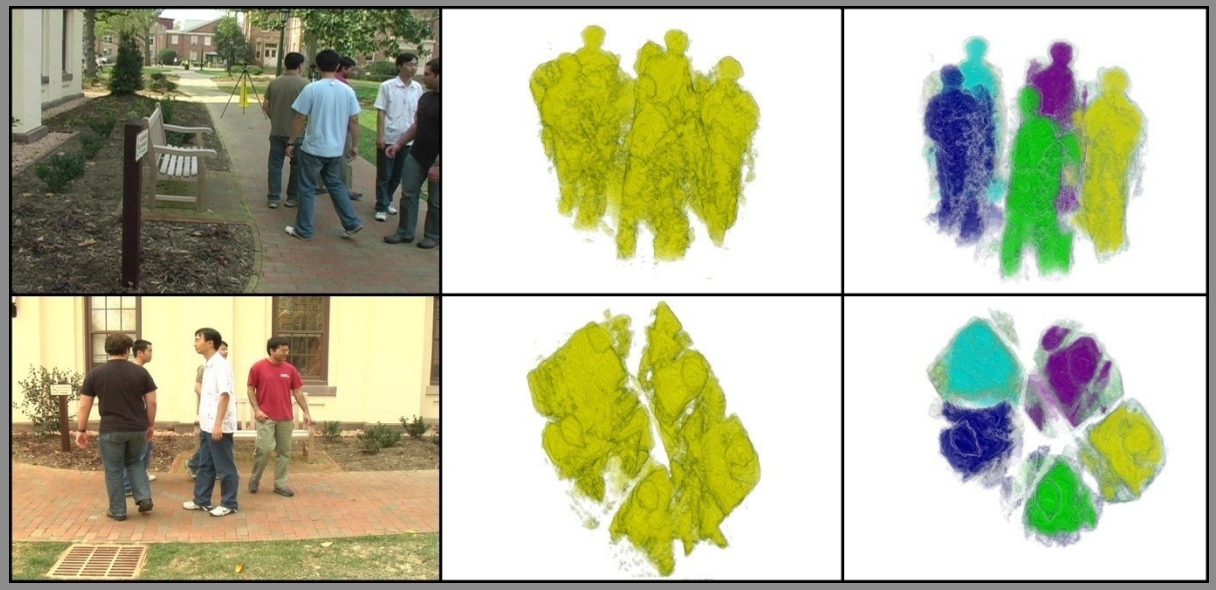
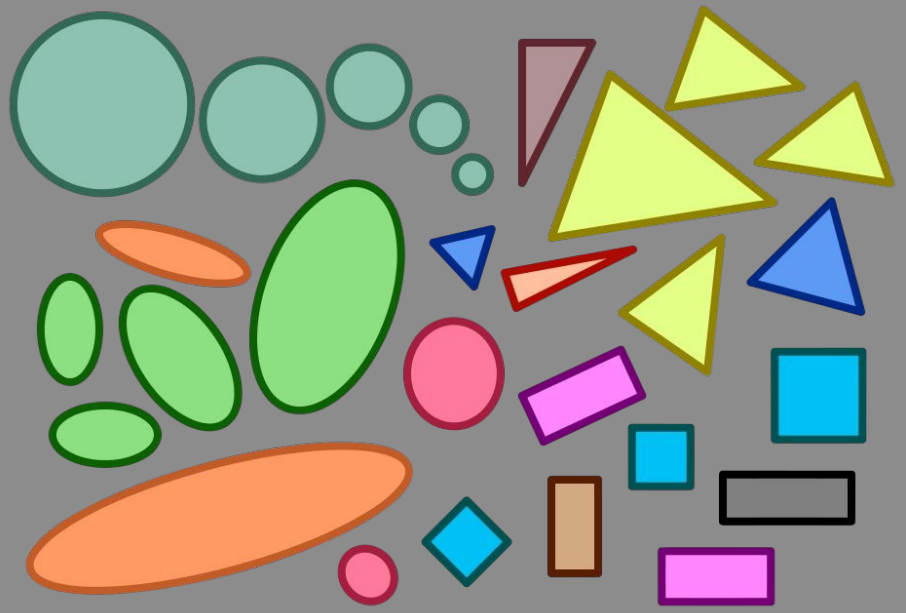


Image Segmentation

- By dividing the whole image into regions which correspond to some uniformity/similarity
- Uniformity measures are
  - Intensity
  - Color parameters
  - Boundary
  - Shape
  - Texture
  - Motion



Input Image

Segmentation Maps

Goal of Segmentation

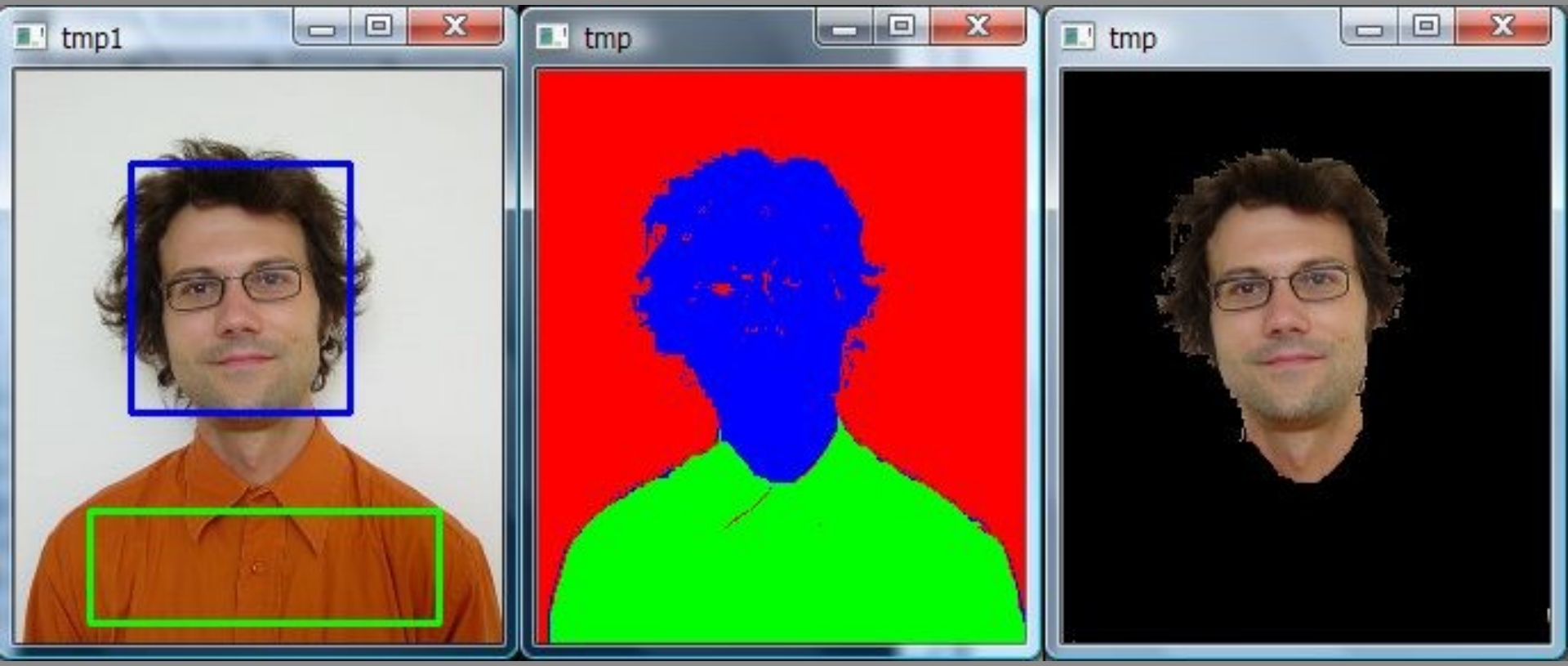


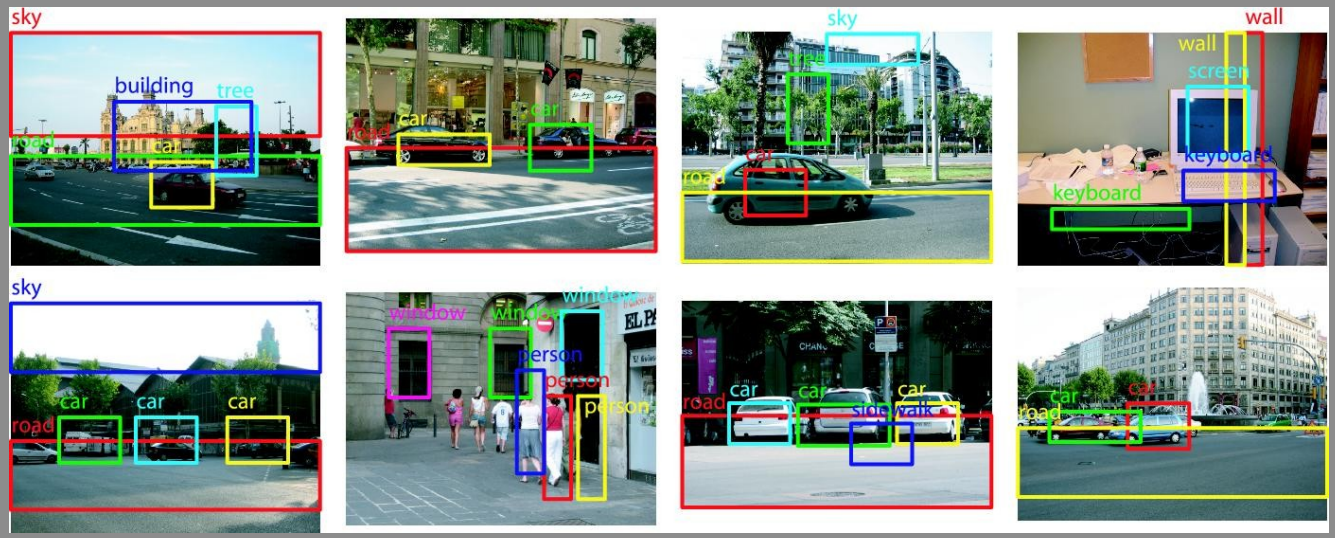
Image Segmentation

Table 10.1 Simple conceptual examples and possible approach to segmentation

Object(s)	Image	Purpose of segmentation	Preferred approach
Aeroplane	Aeroplane in the sky	Tracking	Motion, colour
Human faces	Crowded shopping mall	Face recognition surveillance system	Colour, shape
Man made structures	Aerial satellite photograph	Intelligence acquisition from aircraft	Texture, colour
Cultivated regions	Uncultivated regions	LandSAT survey	Texture, shape
Granny Smiths apples, pears	Various fruits on a conveyor belt	Automated fruit sorting	Colour, shape

Image Segmentation: Connecting Low Processing to High Level Processing

- Link between the low level processing and high le processing
- Low level processing operates on the arrays of ra
- High level processing is concerned with manipula high level abstract data representation



Class Practice Example

Class Practice Problem

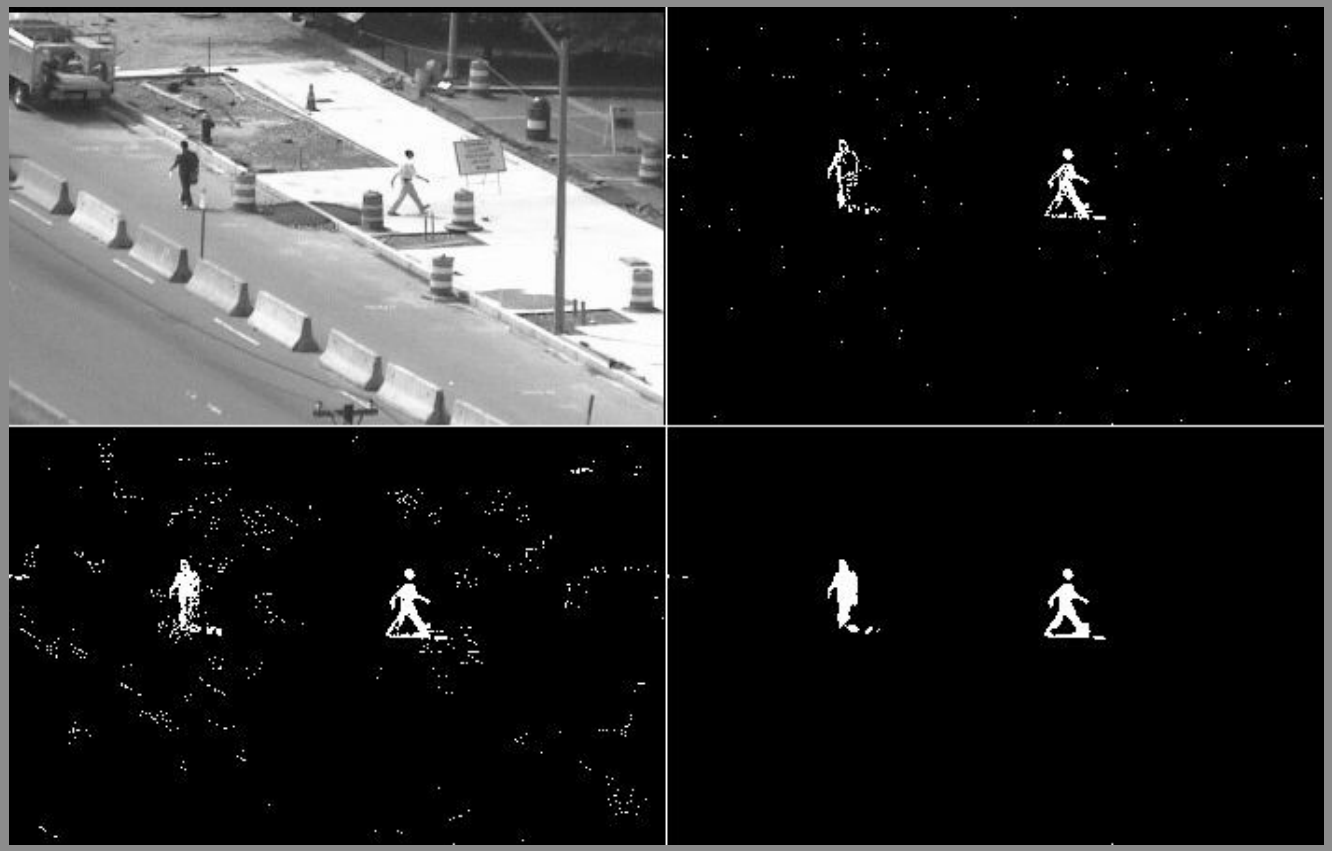
Class Practice Example

Image Segmentation

- Approaches for segmentation of intensity imag
  - threshold techniques,
  - edge-based methods,
  - region-based techniques, and
  - connectivity-preserving relaxation methods

Threshold Technique

- Based on local pixel information



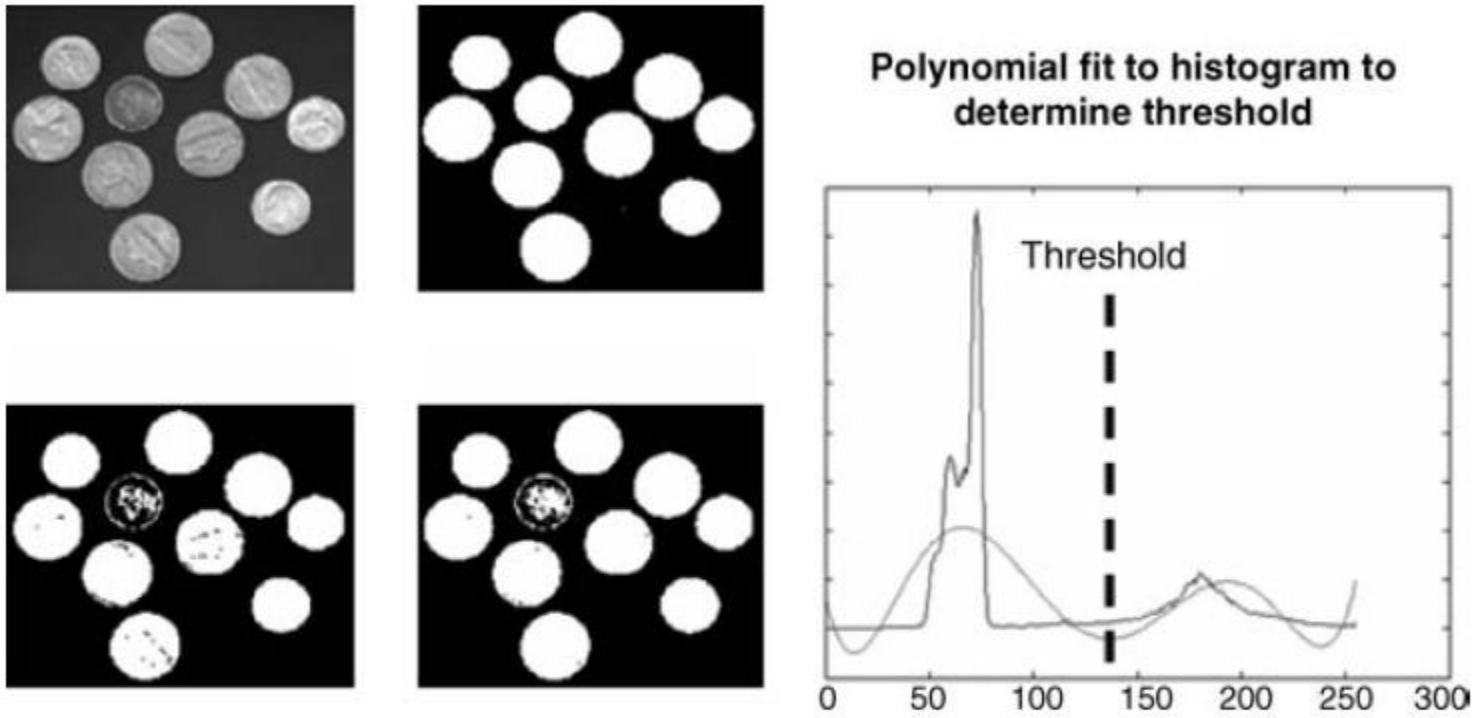


Figure 10.1 Top left: original image. Top right: result after manual selection of threshold. Bottom left: result of automatic threshold selection by polynomial fitting to image histogram. Bottom right: result of automatic threshold selection using Otsu's method. The image on the right shows the histogram and the result of fitting a sixth-order polynomial

THRESHOLDING

- One of the widely methods used for image segmenta
  useful in discriminating foreground from the bac
  selecting an adequate threshold value T, the gray
  can be converted to binary image.

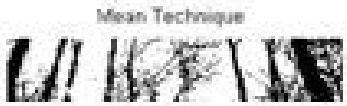
THRESHOLDING TECHNIQUES

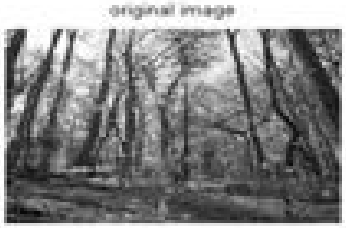
1. **MEAN TECHNIQUE-** This technique used the mean value of the
   the threshold value and works well in strict cases of the i
   approximately half to the pixels belonging to the objects and oth
   background.
2. **P-TILE TECHNIQUE-** Uses knowledge about the area size of the
   object to the threshold an image.
3. **HISTOGRAM DEPENDENT TECHNIQUE (HDT)-** separates t
   homogonous region of the object and background of an image.
4. **EDGE MAXIMIZATION TECHNIQUE (EMT)-** Used when the
   more than one homogenous region in image or where ther
   illumination between the object and its background.
5. **VISUAL TECHNIQUE-** Improve people's ability to accurately se
   target items.

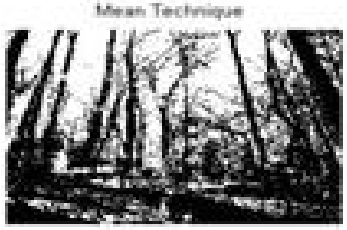
Segmentation Techniques

Fig.1. Original Images

Segmentation Techniques

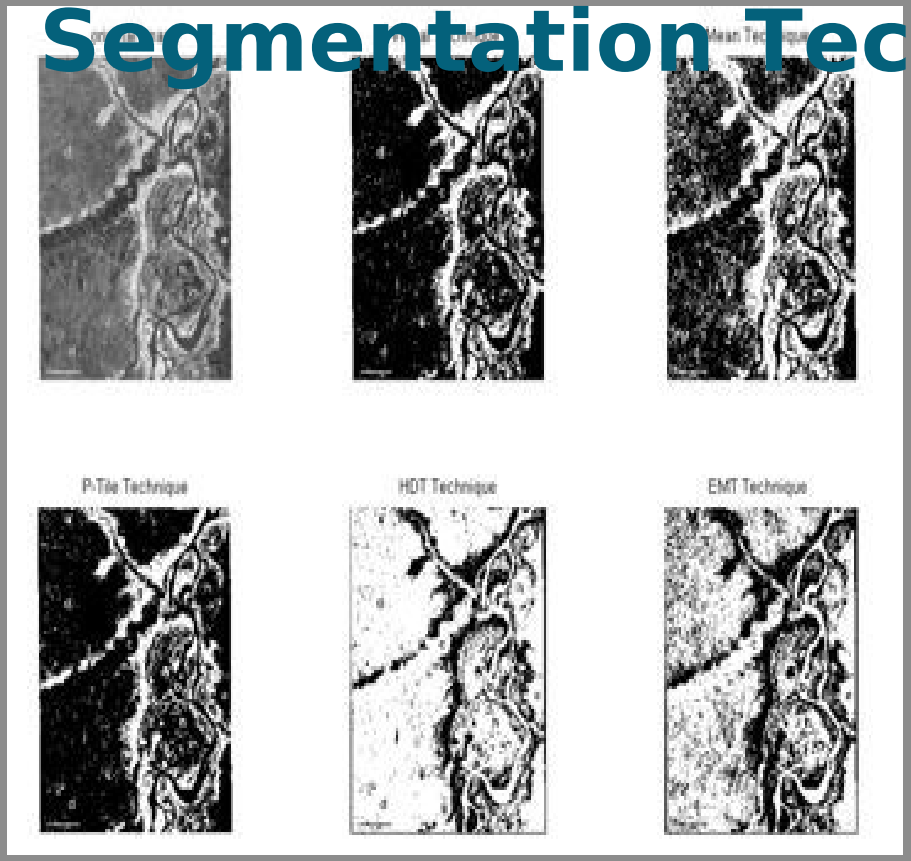






Threshold techniques from left to right original image, Vis technique T = 127, Mean Technique, P-Tile technique T = 127, I Technique and EMT Technique

Segmentation Techniques



T = 167

T = 43

Problems with thresholding

- There are several serious limitations to simple glo thresholding:
  - there is no guarantee that the thresholded pixels w contiguous (thresholding does not consider the spa relationships between pixels);
  - it is sensitive to accidental and uncontrolled variati illumination field;
  - it is only really applicable to those simple cases in v entire image is divisible into a foreground of object similar intensity and a background of distinct inten objects.

Edge-Based Technique

- Edge-based methods center around contour detec

Region-Based Technique

- Partitionedinto connectedregions by grouping neighboring pixels of similar intensity levels.
- Adjacent regions are then merged under some involving perhaps homogeneity or sharpness o boundaries.

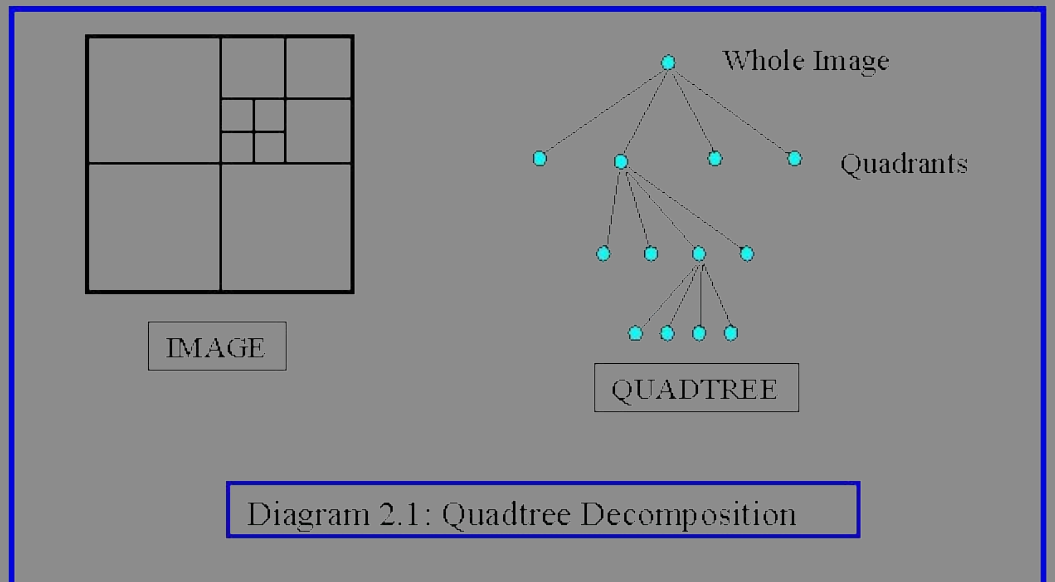


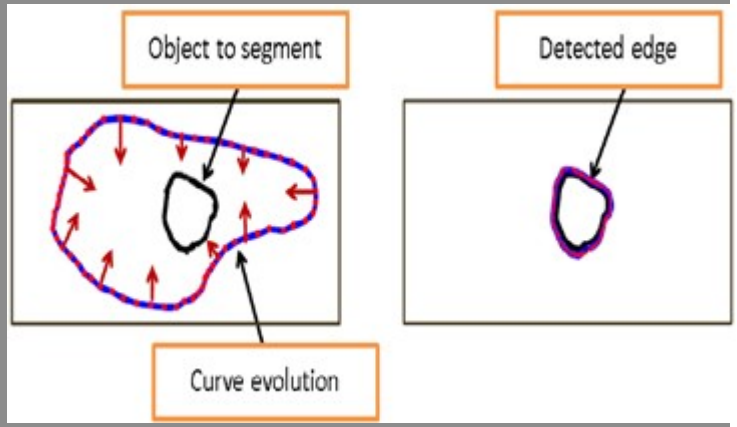
Diagram 2.1: Quadtree Decomposition

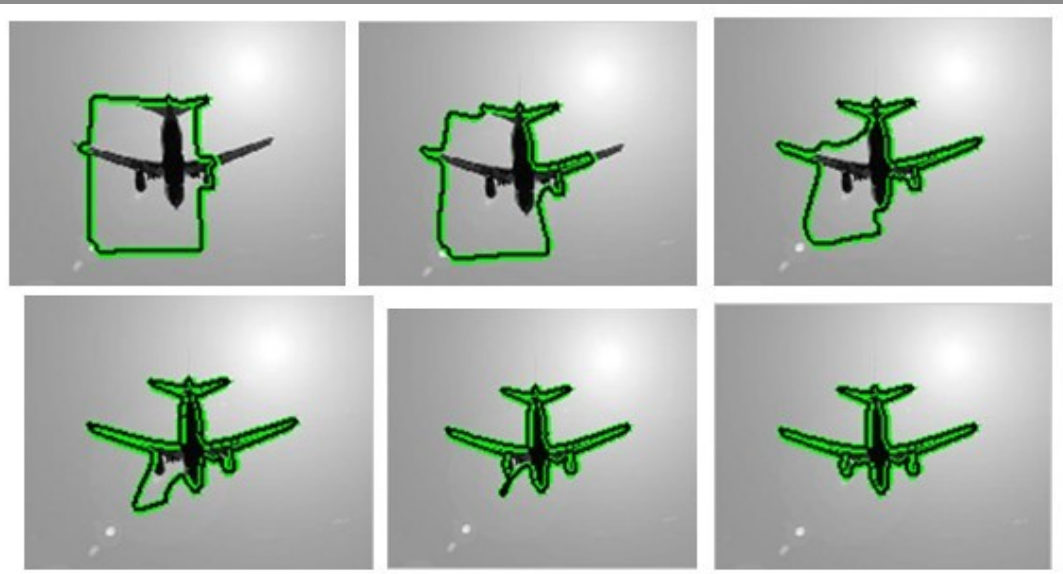
Edge-based Segmentation

- Emphasis:
  - Determine the boundaries that separate regions
- Common approaches
  - Find edge points in the image
    - Gradient based methods
    - Second order methods
  - Linking these points in some way to produce descri
    edges in terms of lines, curves etc

Connectivity-Preserving Relaxation-Based Segmentation (Active Contour Model or Sn

- Also referred as *active contour model or snakes*
- Begin with initial boundary shape represented in of spline curves, and iteratively modify it by apply various shrink/expansion operations according to energy function





Snakes

- Contour possesses an energy (E_{snake}) which is defined as the sum of the three energy terms.

$$E_{snake} = E_{\text{int}ernal} + E_{external} + E_{constra\text{int}}$$

where $E_{internal}$ represents the internal energy of the spline due to ben $E_{external}$ denotes image forces, and $E_{constraint}$ denotes external constraint forces.

- The energy terms are defined such that the final position of th contour will have a minimum energy (E_{min})

- Therefore the problem of detecting objects reduces to an ene minimization problem.

Framework for snakes

- A higher level process or a user initializes any curve *close to the object boundary.*
- The snake then starts *deforming* and moving towards the desired object boundary.
- In the end it completely “wraps” around the object.

(Digram courtesy “Snakes, shapes, gradient vector flow”, Xu, Princ

Problems in segmentation

- Most commercial vision systems use region-based segmentation schemes bas pixel-intensity values.
- These segmentation techniques assume objects of interest possess uniform shad that a significant and constant gray-leve change occurs between the object(s) of i and the background.

Problems in segmentation

- However, in many vision applications, these assumptions have p erroneous. Therefore, these techniques are considered fragile a require controlled conditions or human supervision.
- Effects of
  - uneven sample illumination,
  - shadowing,
  - partial occlusion,
  - clutter,
  - noise, and
  - subtle object-to-background changes can all contribute to errors in b segmentation processes.

Problems in Segmentation: Application Related Tolerance Level

- Some applications, such as robotic vision, can a few errors in the segmentation process, the objects are classified correctly
- In other applications, such as clinical analy feature measurements might be critically i and may dictate the use of a sophisticated and computationally expensive segmentation algor

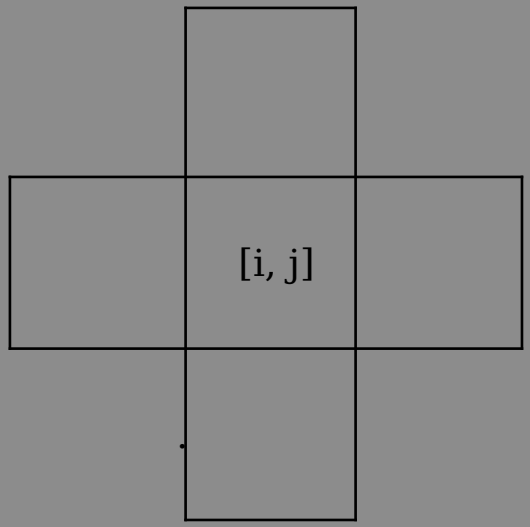
Boundary Detection

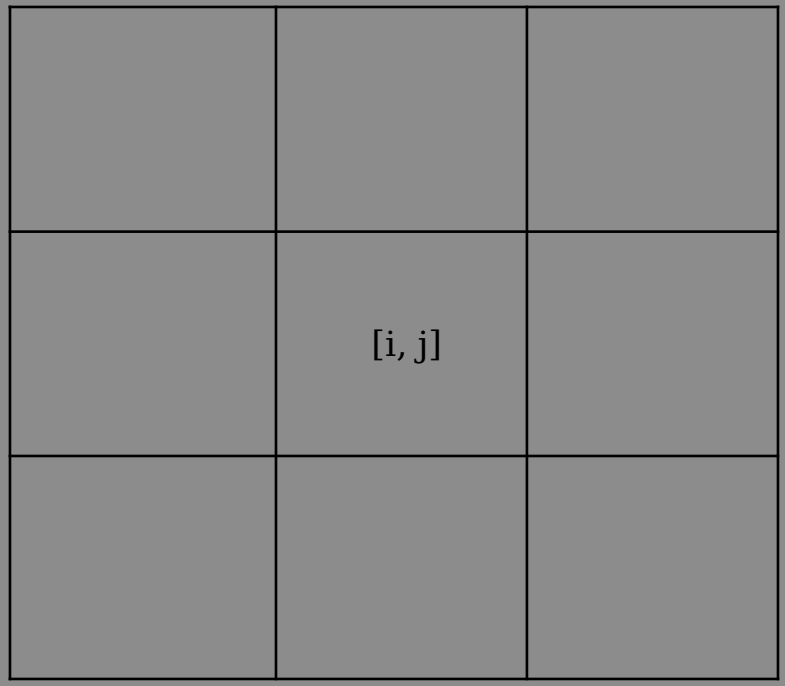
- Human extensively use the boundary information to in
and recognize the object.
- When the various edge operators described above are
to an image they detect local edge elements, or edges
- However an image of disconnected edge elements as
uninformative and additional processing is needed to
these elements into straight and curved boundaries.
- These closed boundaries provide patterns and structu
are more appropriate for the recognition and classific
any objects in the scene.

Boundary Detection

- The goal of boundary formation, or edge linking, is to coherent one dimensional edge feature from many ind local edge elements.
- Boundary formation techniques have to handle instan edge elements do not correspond to any meaningful s boundary, and conversely they have to cope when no elements are present even though a boundary should
- The methods vary in the amount of *a priori* knowledge incorporated into the grouping operation which links edge elements.

4-neighbors & 8 neighbors

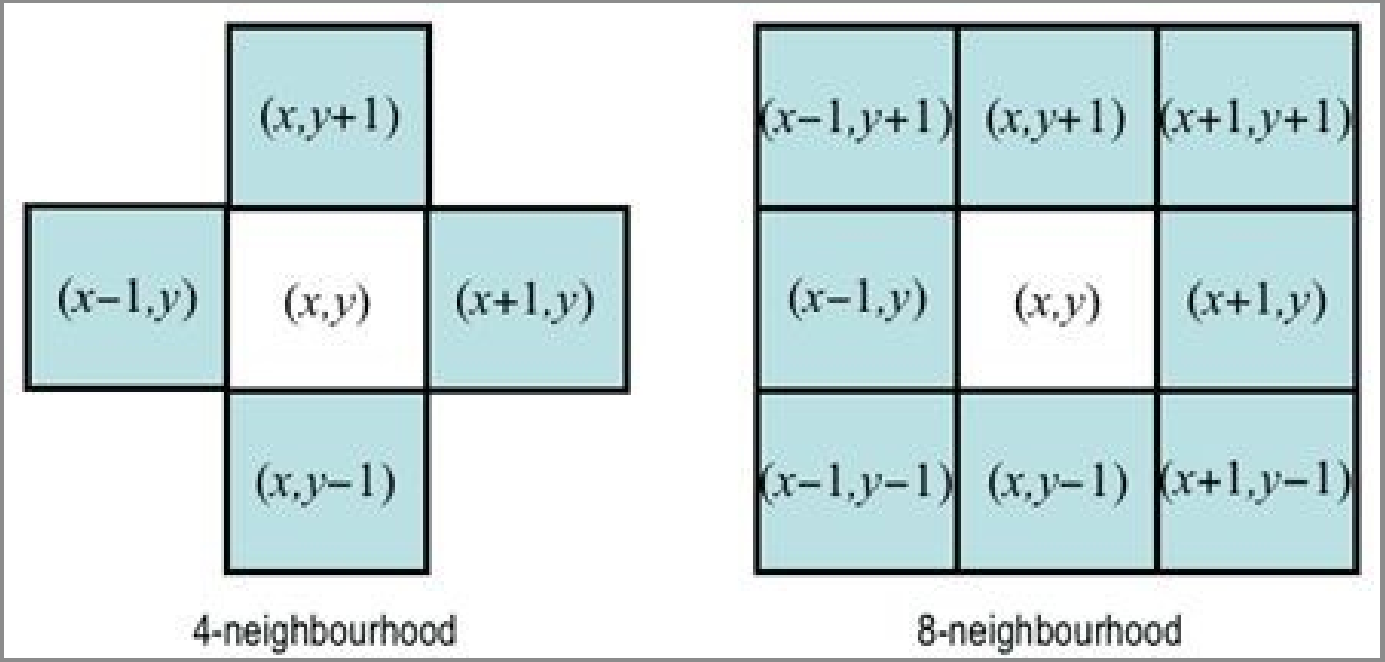




4-neighbors [i+1, j], [i-1, j], [i, j+1], [i, j-1]

8-neighbors [i + 1,j + 1], [i + 1,j - 1], [i - 1,j + 1], [i - 1,j - 1] plus all of the 4- neighbors

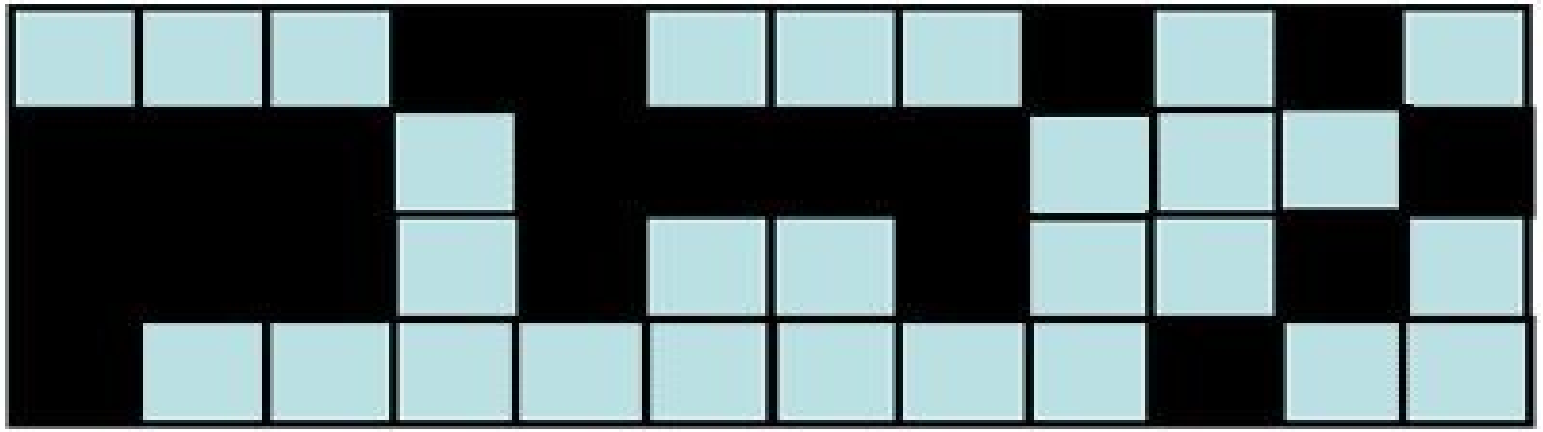
4-neighbors & 8 neighbors



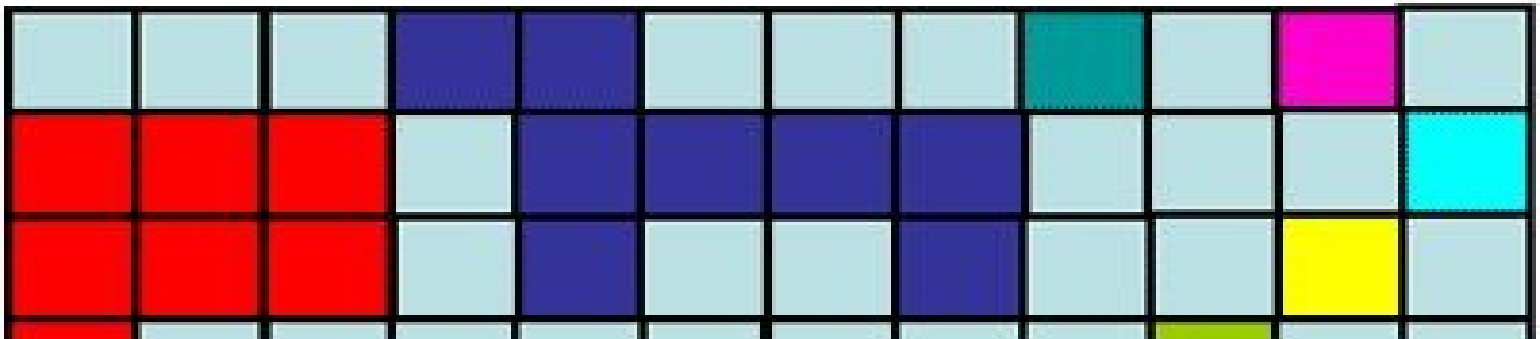
4-neighbors [i+1, j], [i-1, j], [i, j+1], [i, j-1]

8-neighbors [i + 1,j + 1], [i + 1,j - 1], [i - 1,j + 1], [i - 1,j - 1] plus all of the 4- neighbors

Segmentation-Binary

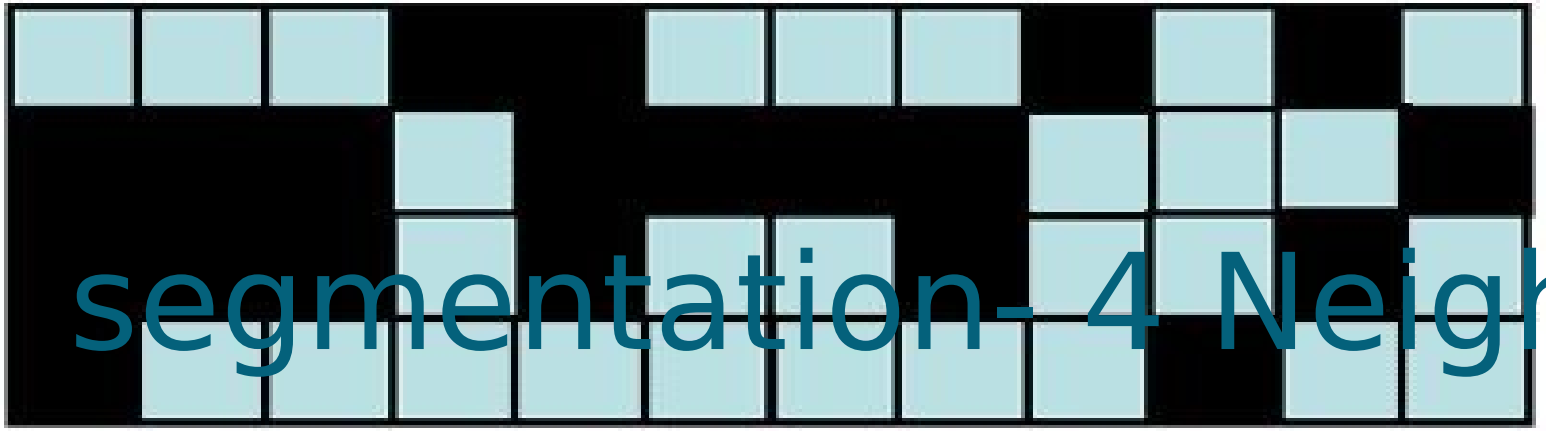


Binary image:
0 - objects;
1 -background

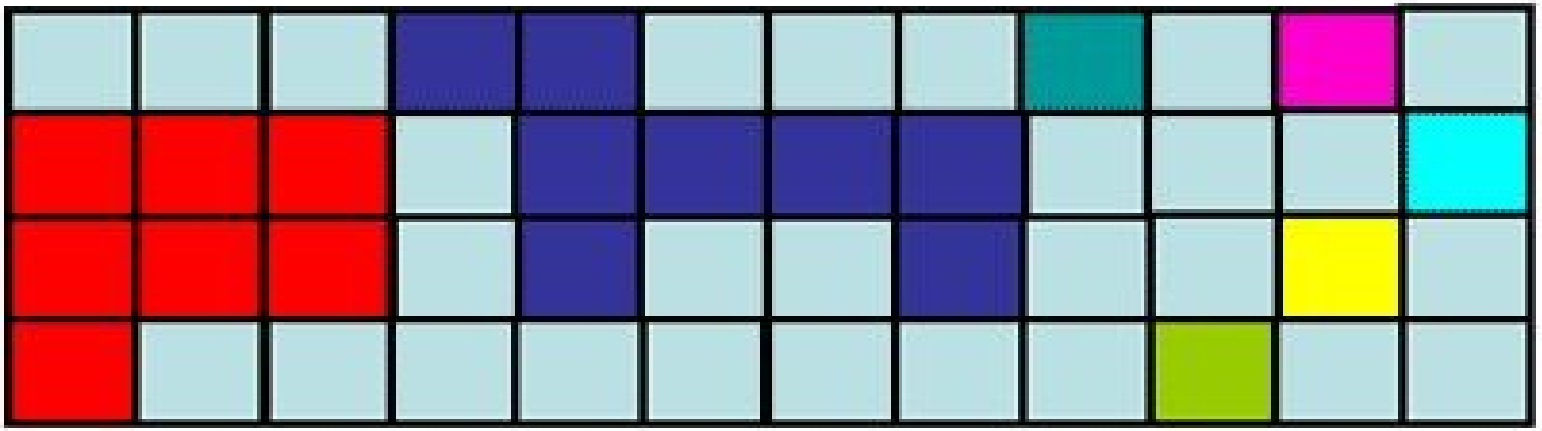


4-connected
objects +

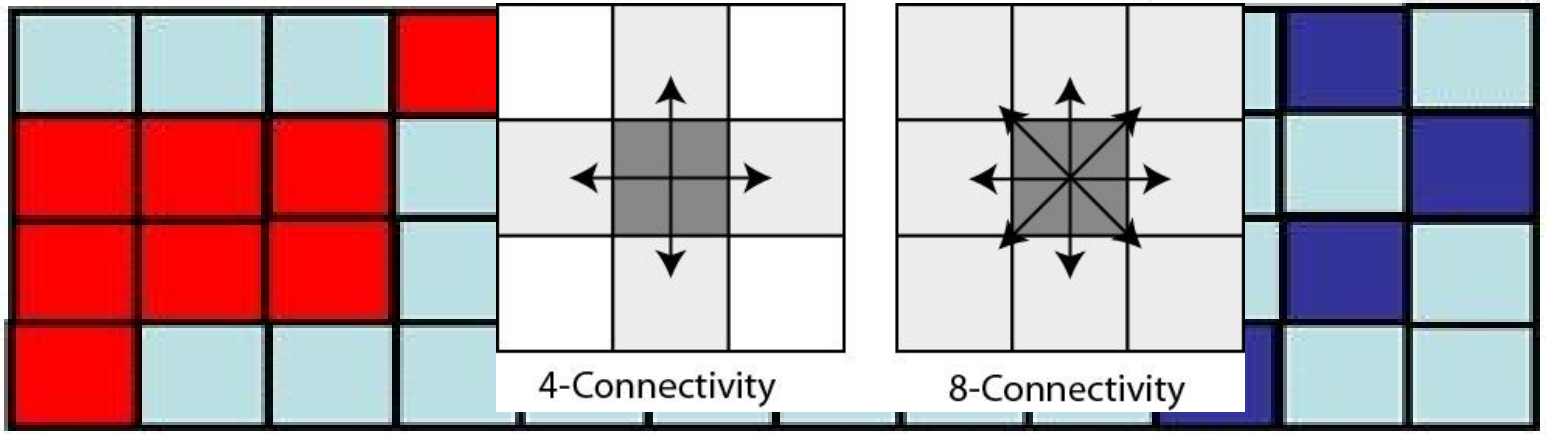
segmentation- 4 Neighborh



Binary image:
0 -objects;
1 -background



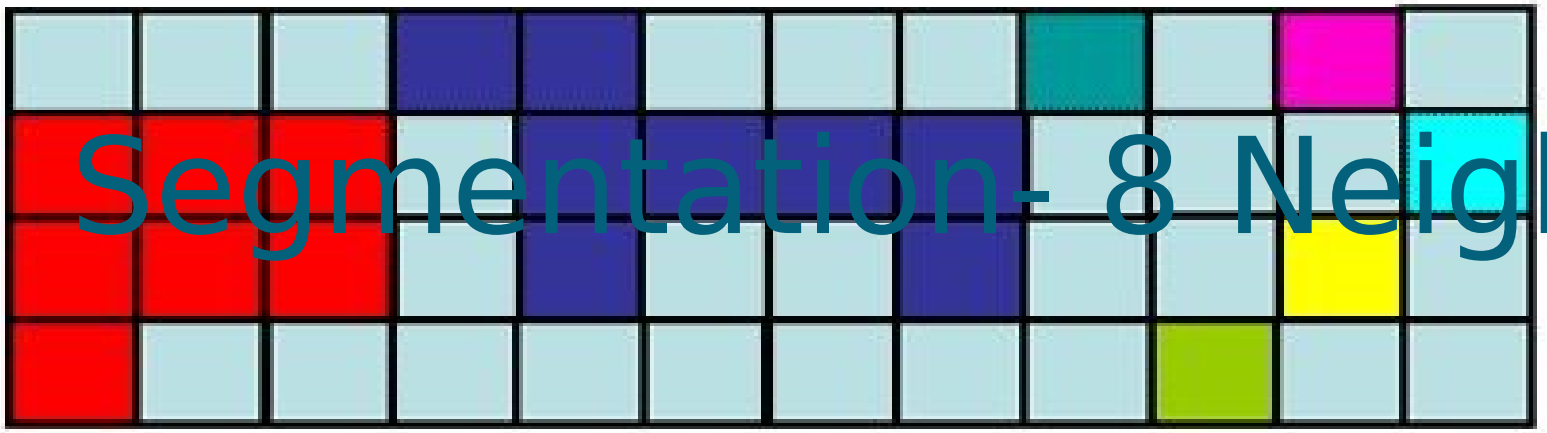
4-connected
objects +
8-connected
background



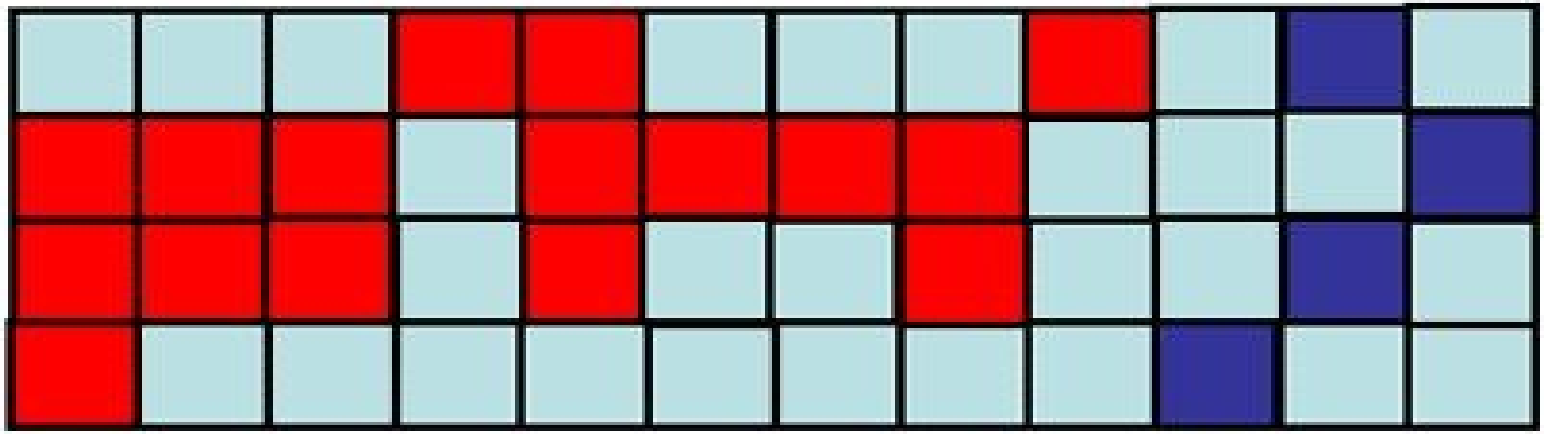
8-connected
objects +
8-connected
background

Segmentation- 8 Neighborh

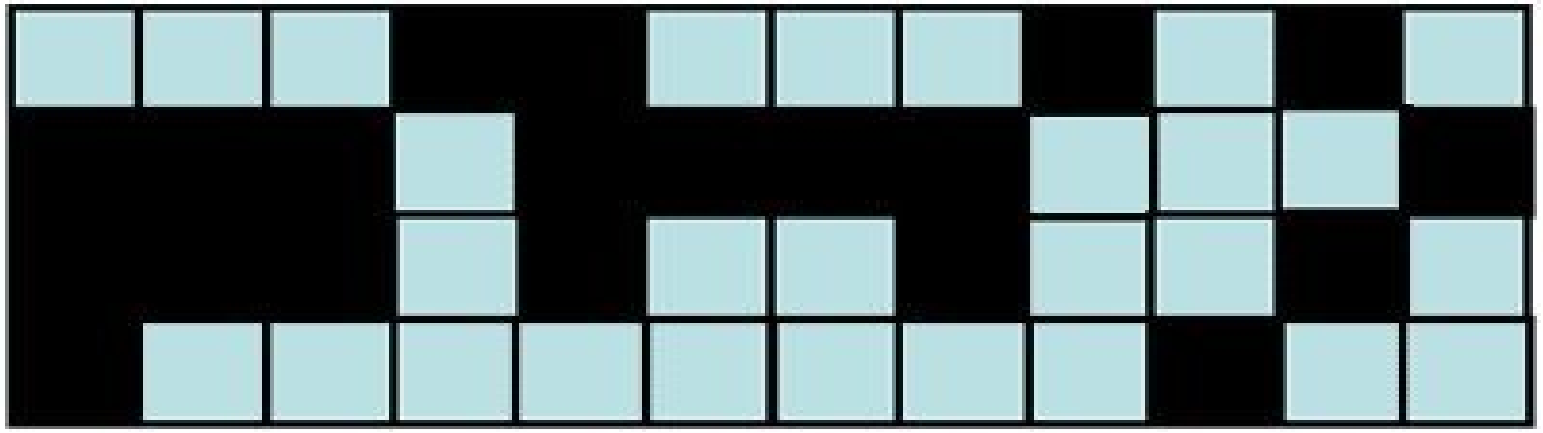
1 -background



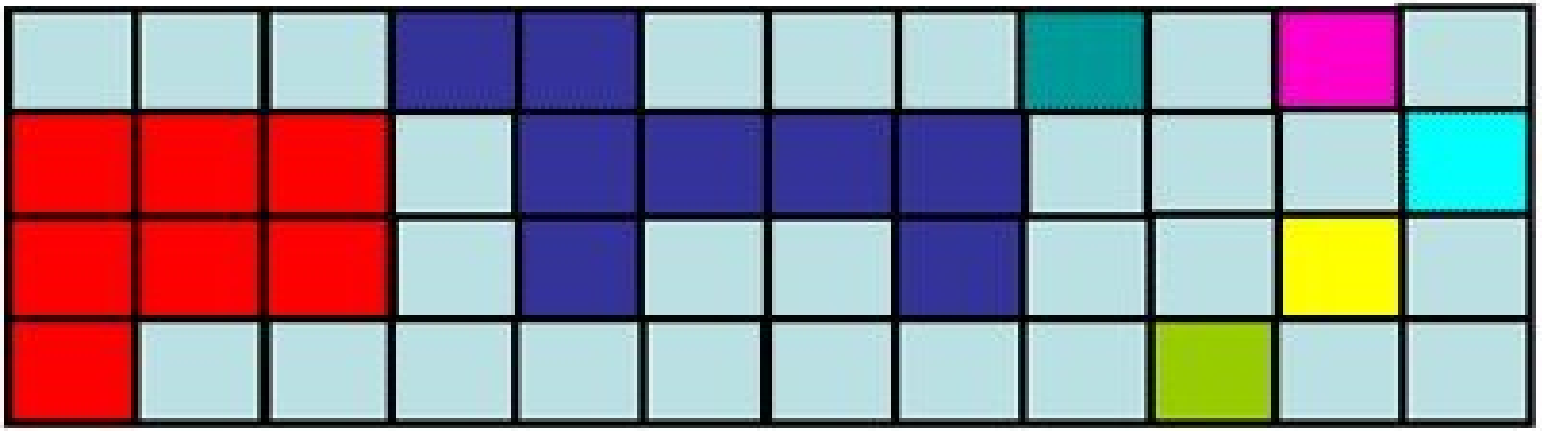
4-connected
objects +
8-connected
background



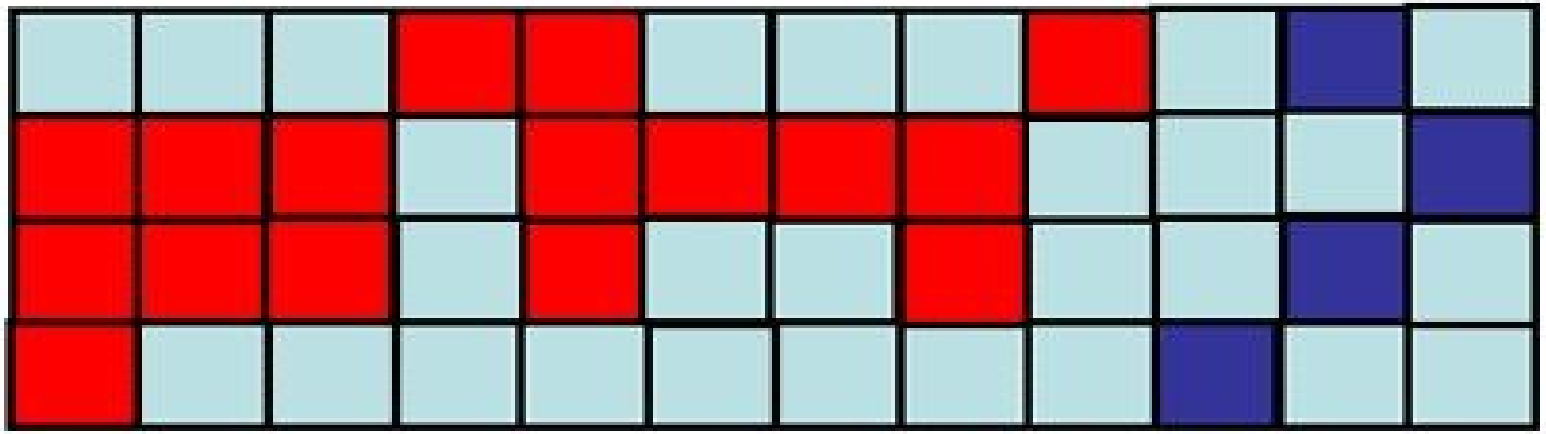
8-connected
objects +
8-connected
background



Binary image:
0 - objects;
1 -background



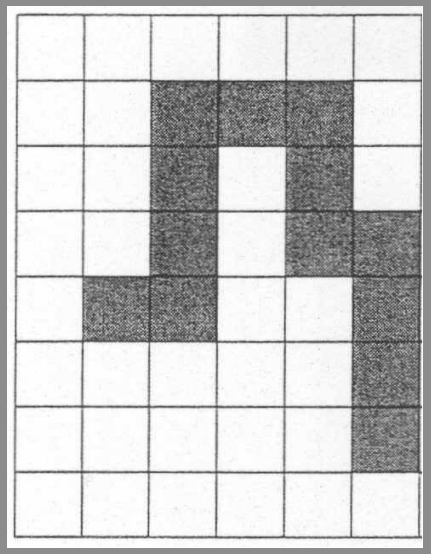
4-connected
objects +
8-connected
background

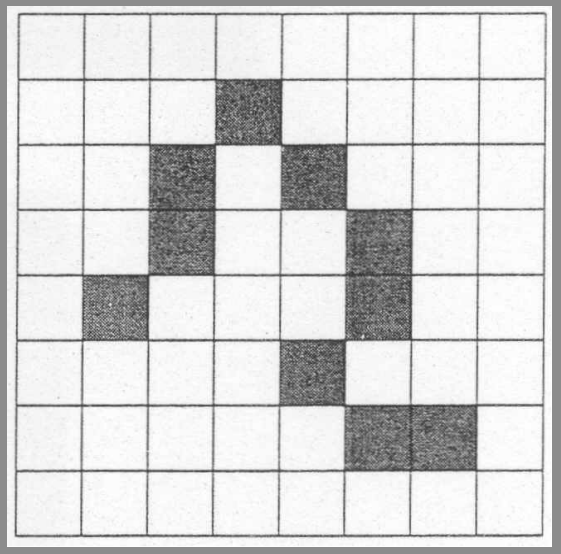


8-connected
objects +
8-connected
background

Path

- A path from the pixel at $[i_0, j_0]$ into the pixel at $[i_n, j_n]$ is a sequence of pixel in dices $[i_0, j_0]$, $[i_1, j_1]$, $[i_2, j_2]$,..., $[i_n, j_n]$ such that the pixel at $[i_{k+1}, j_{k+1}]$ is a neigh bor of the pixel at $[i_k, j_k]$ for all k with $0 < k < n - 1$.
- If the neighbor relation uses 4-connection, then the path is 8-connection, the path is an 8-path.





Foreground & Connectivity

- **Foreground**

  The set of all 1 pixels in an image is called the foreground and is denoted by S.

- **Connectivity**
- A pixel p∈S is said to be connected to q ∈S if there is from p to q consisting entirely of pixels of S.
- For any three pixels p, q, and r in S, we have the follo properties:
  1. Pixel p is connected to p (reflexivity).
  2. If p is connected to q, then q is connected to p (commutativi
  3. If p is connected to q and q is connected to r, then p is conne (transitivity).

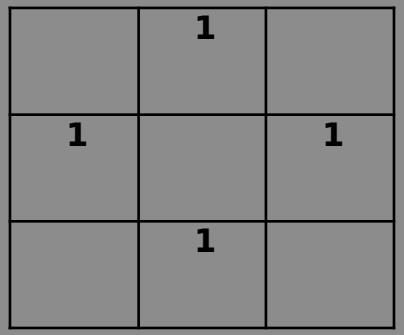
Connected Components & Backgro

- **Connected Components**

  A set of pixels in which each pixel is connect other pixels is called a connected component.

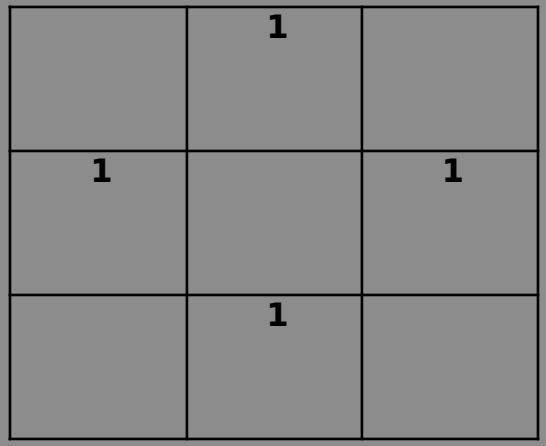
- **Background**

  The set of all connected components of Â (the complement of A) that have points on the bo image is called the background. All other compon Â are called holes.



Connected Components

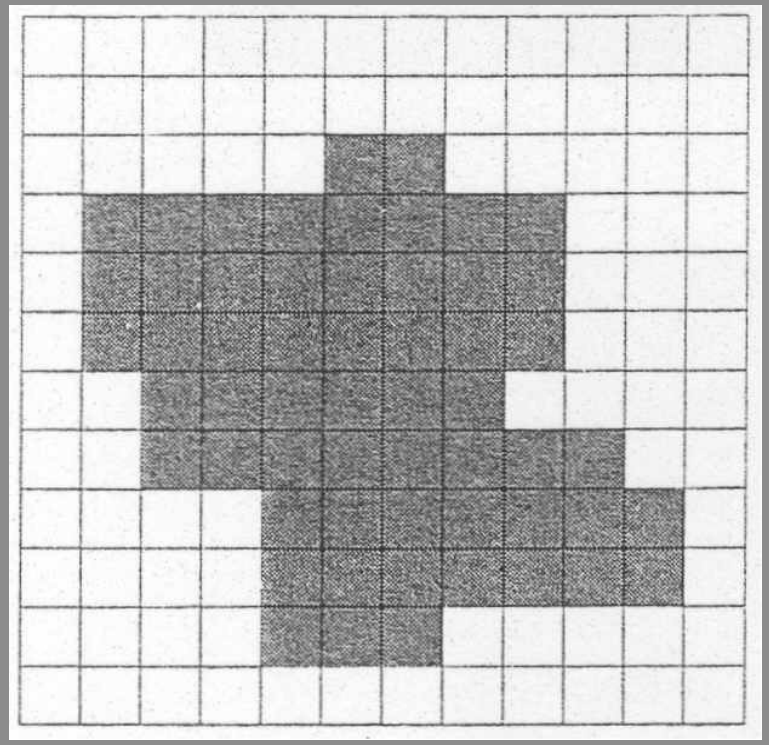
- How many objects and how many holes are in this figu
- If we consider 4-connectedness for both foreground and background, there are four objects that are 1 pixel in there is one hole.
- If we use 8-connectedness, then there is one object ar

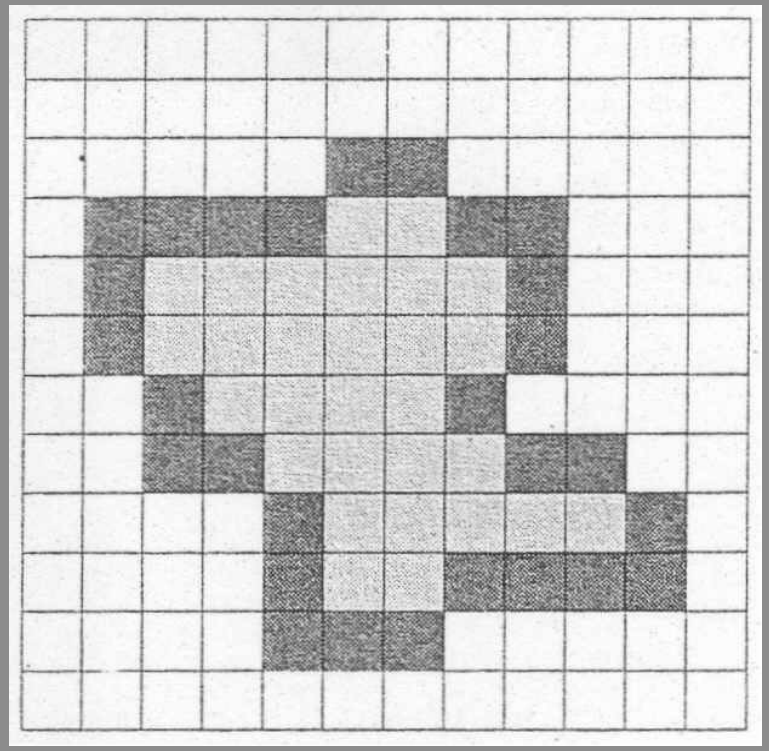


Boundary, Interior & Surrou

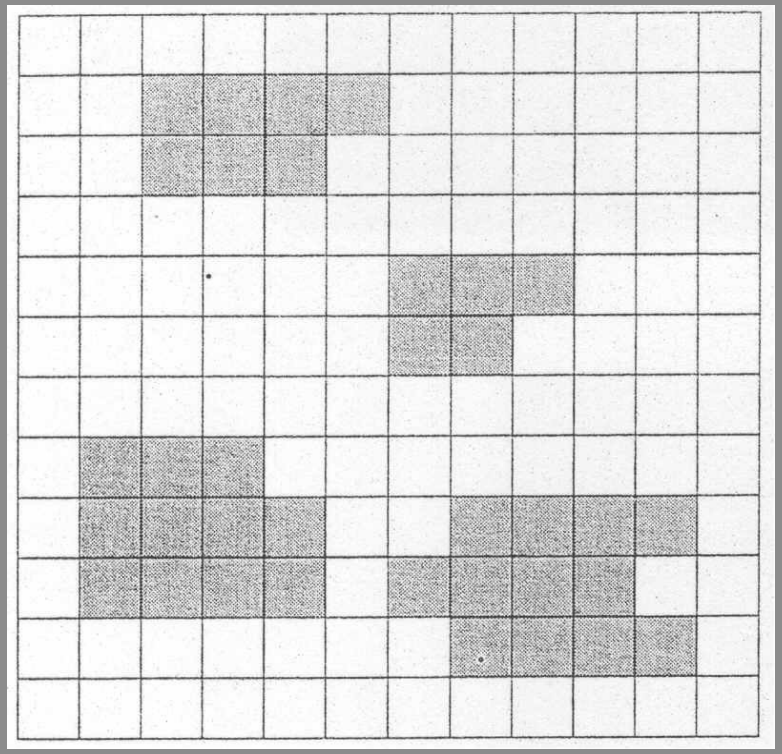
- The boundary of S is the set of pixels of S that hav
  4-neighbors in . The boundary is usually denoted
- The interior is the set of pixels of S that are not in
  boundary. The interior of S is (S - S').
- Region T surrounds region S (or S is inside T), if a
  4-path from any point of S to the border of the pic
  must intersect T.
- Figure NEXT shows an example of a simple binary
  with its boundary, interior, and surrounds.

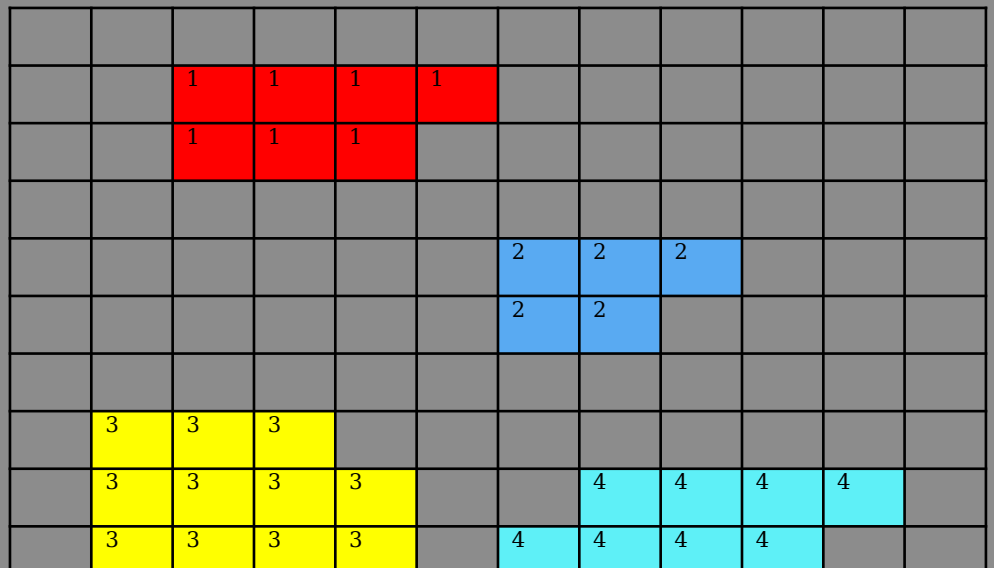
Boundary, Interior & Surrou





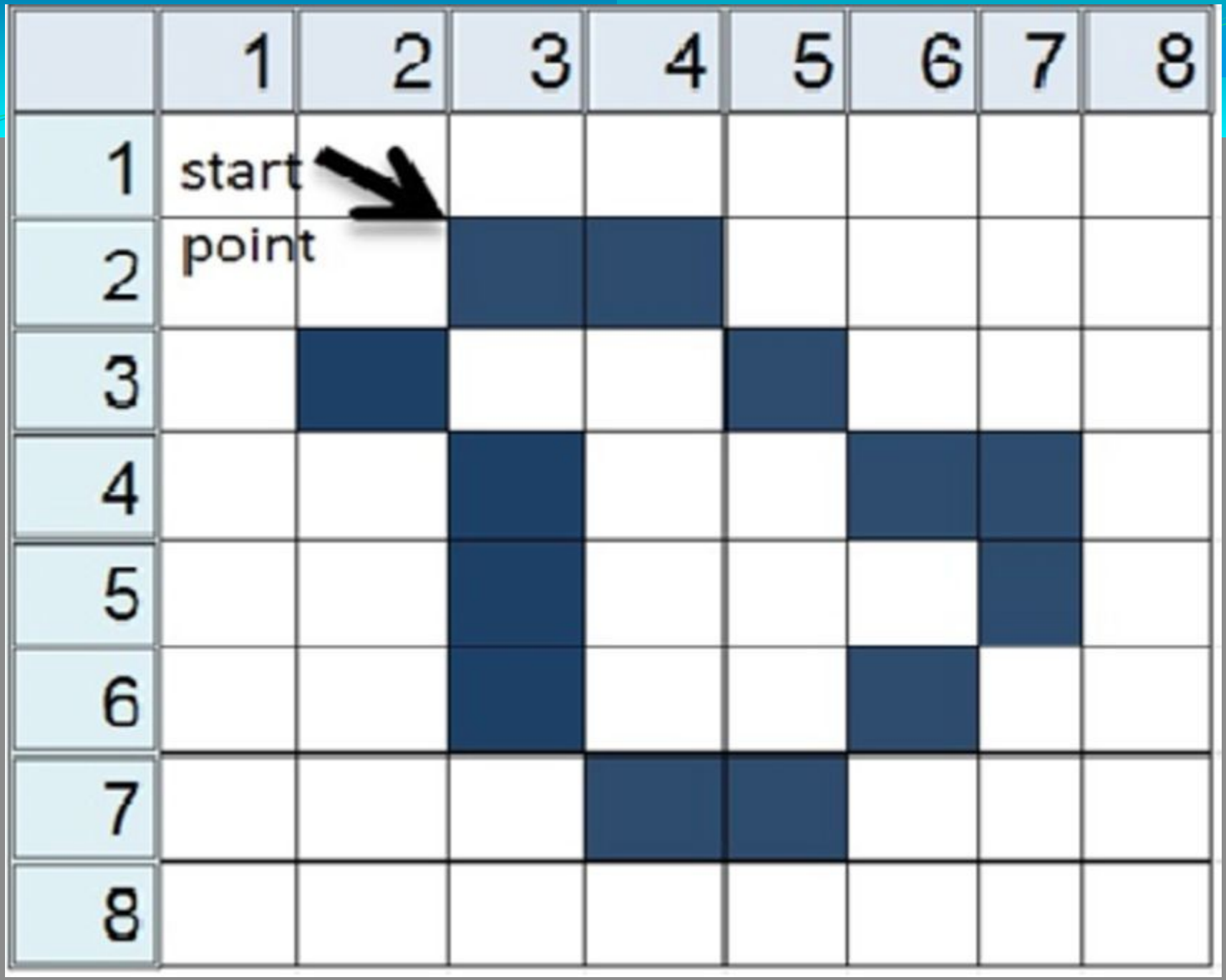
Component Labeling





Run Length Encoding

- One compact representation of a binary image run-length encoding.
- In this representation, numbers indicating the of the runs of 1 pixels in the image are used to represent the image.
- This coding has been used for image transmiss
- Additionally, some properties, such as the area objects, may be directly computed from their run-length codes.


1 2 3 4 5 6 7 8
1
2
3
4
5
6
7
8
start
point

Run Length Encoding

- Two approaches are commonly used in run-length enc
- In the first, the start position and lengths of runs of 1s from each row are used.
- The other approach uses only lengths of runs, starting with the length of the 1 run.
- Area of an object is given as (where m_i is the number of runs in the i^{th} row):

$$A = \sum_{i=0}^{n-1} \sum_{k=0}^{\frac{m_i - 1}{2}} r_{i,2k+1}$$

Run-length Encoding

- This encoding method is frequently applied to
(or pixels in a scan line). It is a small co
component used in JPEG compression.
- In this instance, sequences of image elements $X_1, X_2, \ldots, X_n$ are mapped to pairs $(c_1, l_1), (c_2, l_2), \ldots, (c_n, l_n)$ where c_i represent image intensity or colour a
l_i the length of the ith run of pixels

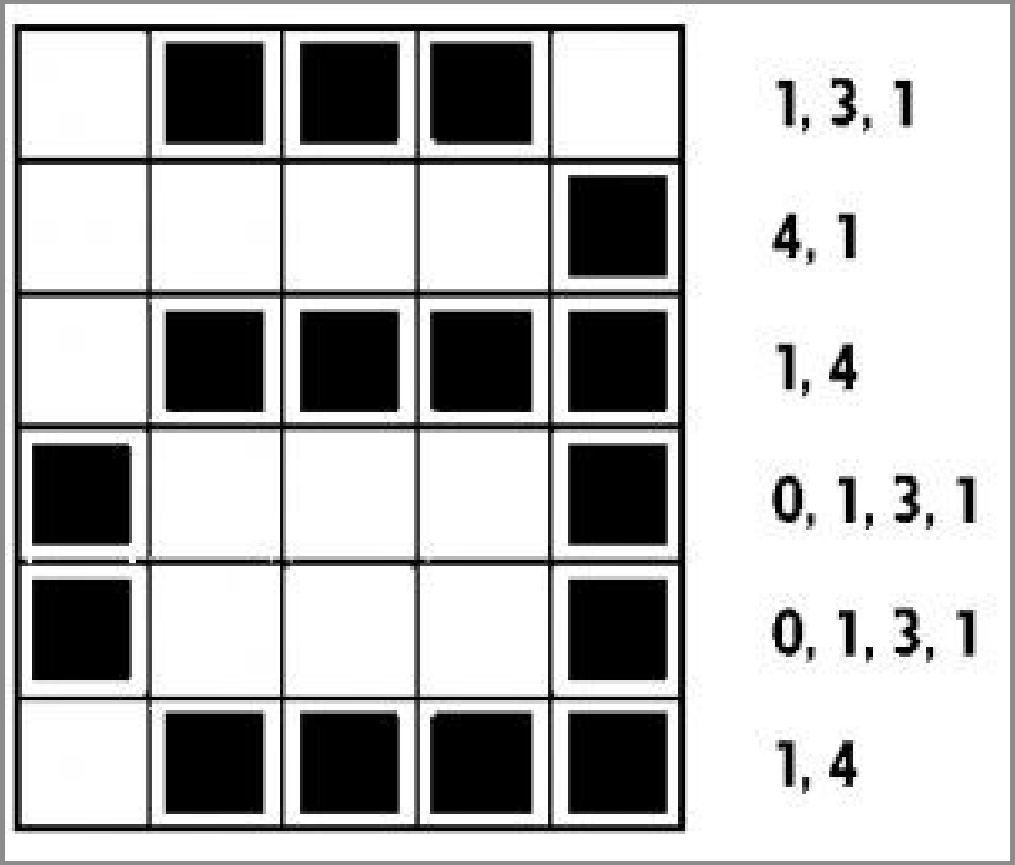
Run Length Encoding

1	1	1	0	0	0	1	1	0	0	0	1	1	1	1	0	1	1	0	1	1	1
0	0	0	0	.1	1	1	1	1	1	1	1	1	1	1	1	1	0	1	1	1	1
1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1

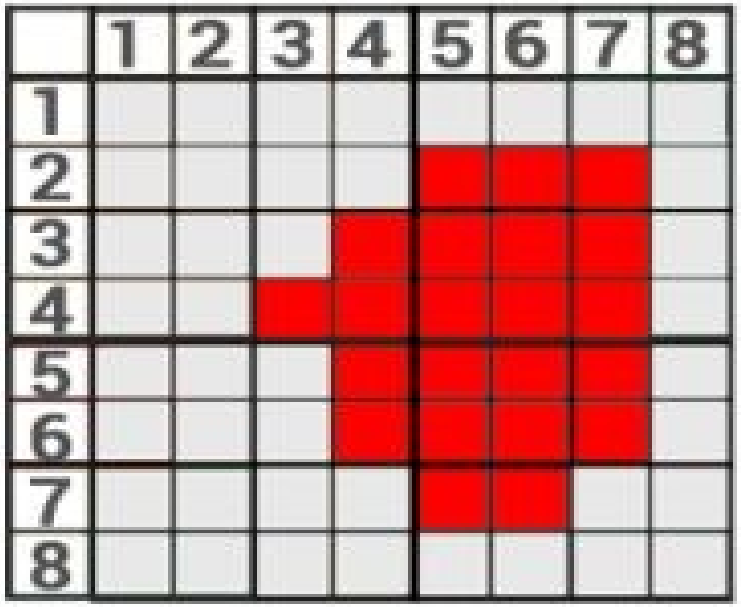
Start and length of 1 runs: (1, 3) (7, 2) (12, 4) (17, 2) (20, 3)
(5, 13) (19, 4)
(1, 3) (17, 6)

Length of 1 and 0 runs: 3,3,2,3,4,1,2,1,3
0, 4, 13, 1, 4
3,13,6

Run Length Encoding (Exan



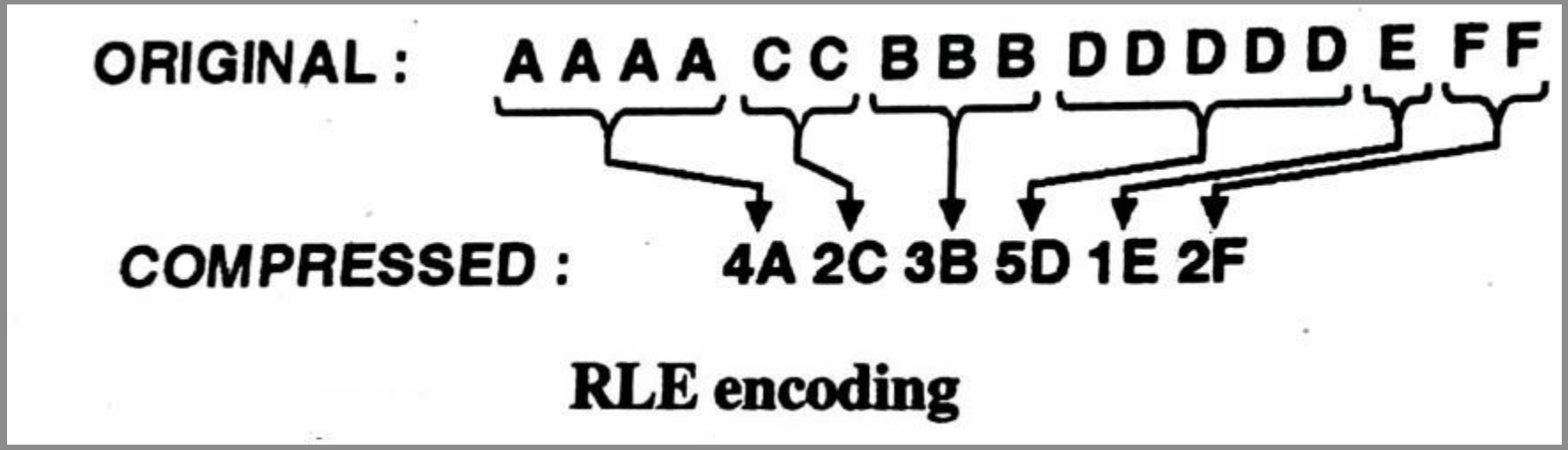
RLE



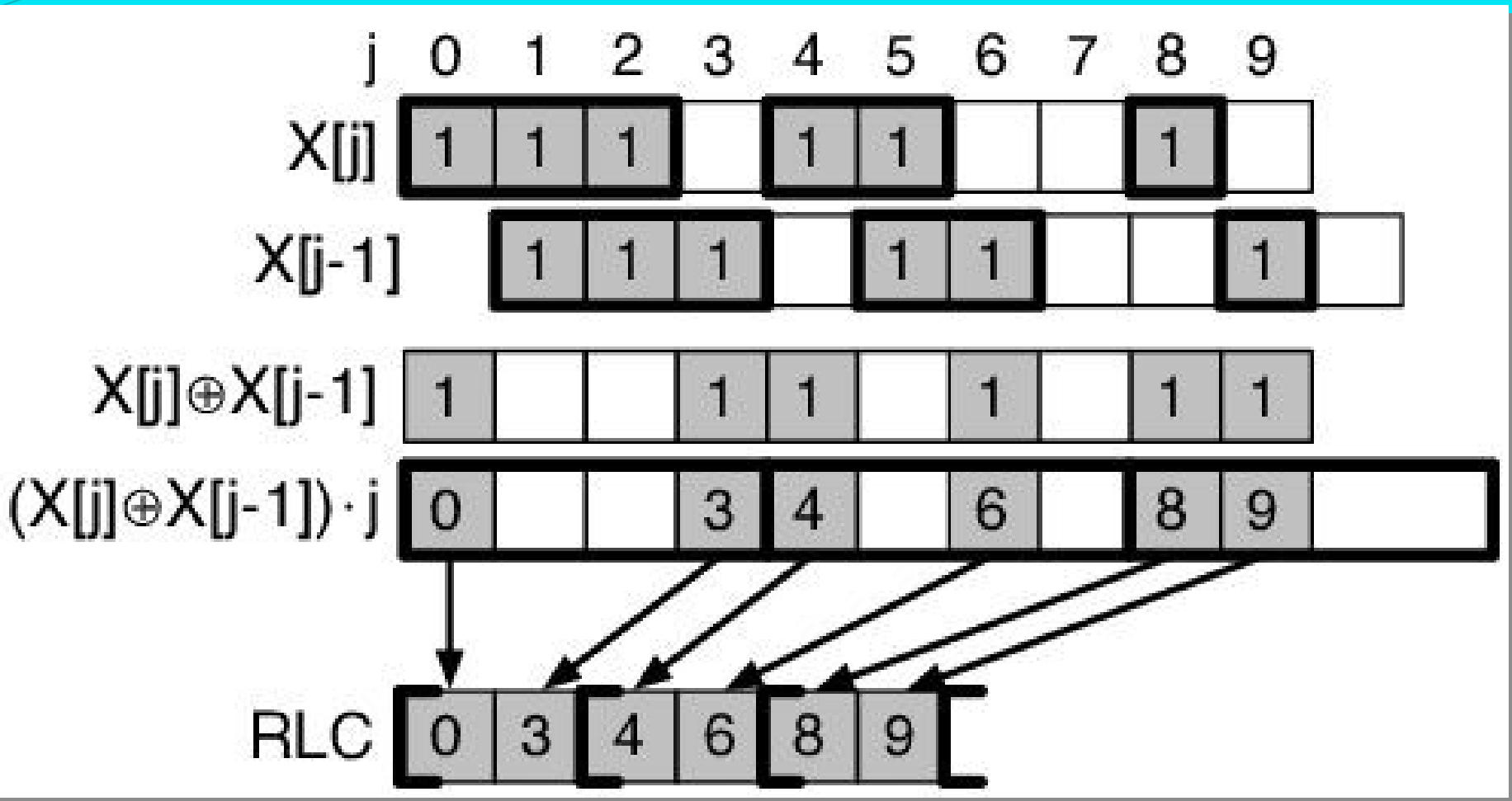
	1	2	3	4	5	6	7	8
1								
2								
3								
4								
5								
6								
7								
8								

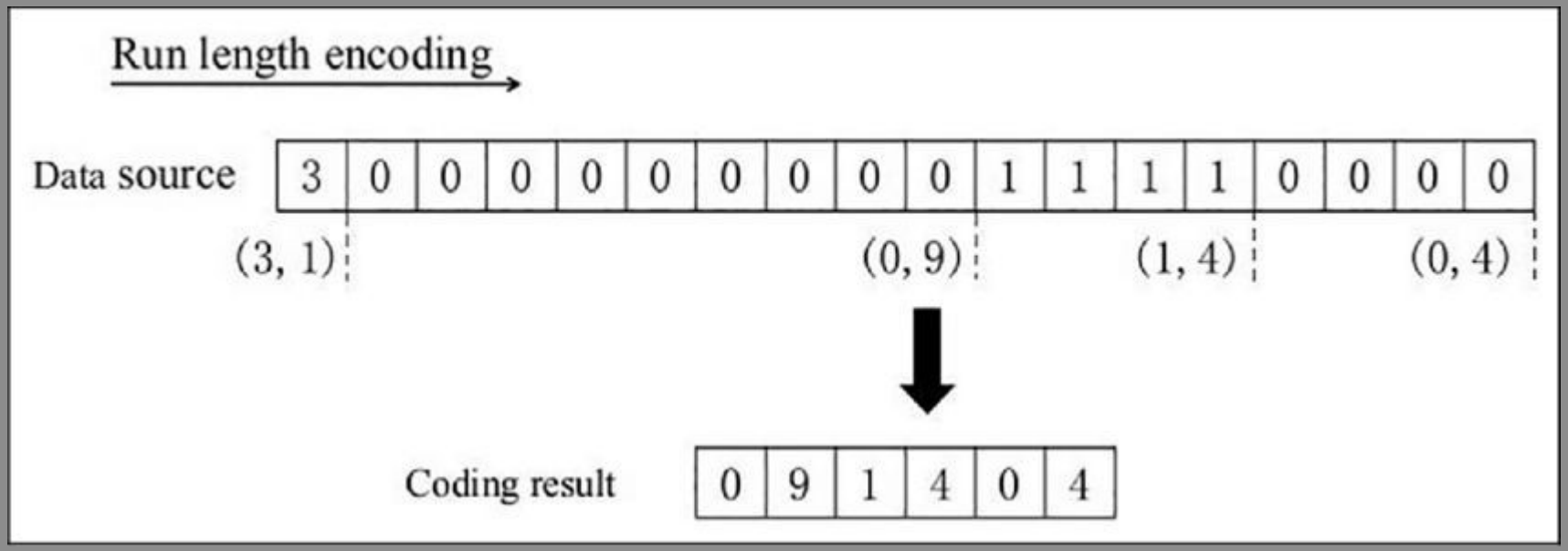
(8, 8, 1)

(0,8)
(0,4) (1,3) (0,1)
(0.3) (1,4) (0,1)
(0,2) (1,5) (0,1)
(0,3) (1,4) (0,1)
(0,3) (1,4) (0,1)
(0,4) (1,2) (0,2)
(0,8)

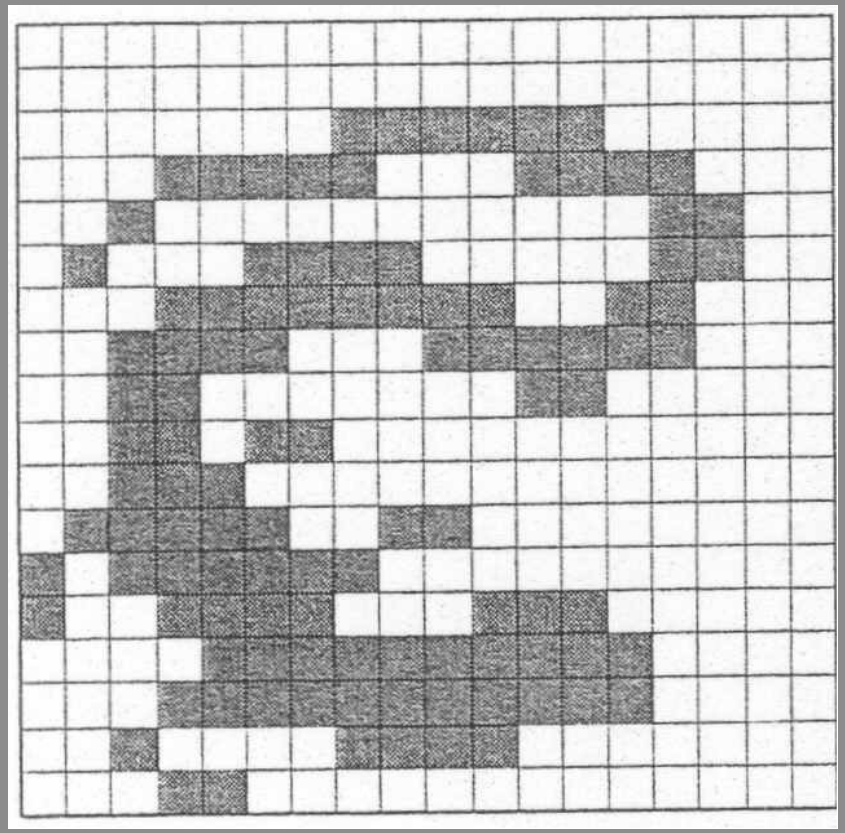


RLE encoding


j 0 1 2 3 4 5 6 7 8 9
X[j] 1 1 1 1 1 1 1
X[j-1] 1 1 1 1 1 1 1
X[j]⊕X[j-1] 1 1 1 1 1 1 1
(X[j]⊕X[j-1])·j 0 3 4 6 8 9
RLC 0 3 4 6 8 9


Run length encoding
Data source
3 0 0 0 0 0 0 0 0 0 1 1 1 1 0 0 0 0
(3, 1)
(0, 9)
(1, 4)
(0, 4)
Coding result
0 9 1 4 0 4

Quiz: Write the run-length Code



Quadtrees

- Quadtrees are both an indexing structure for and compression scheme for binary images
  - A quadtree is a tree where each non-leaf node has f children
  - Each node is labelled either B (black), W (white) or (gray)
  - Leaf nodes can only be B or W

Quadtree Example

1	1	0	1	1	1	0	0	0	1	1	1	1	1	1	1
1	1	0	1	1	1	0	0	0	1	1	1	1	1	1	1
1	1	0	0	0	0	0	0	0	0	1	1	1	1	1	1
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Quadtree Example

1	1	0	1	1	1	0	0	0	1	1	1	1	1	1	1
1	1	0	1	1	1	0	0	0	1	1	1	1	1	1	1
1	1	0	0	0	0	0	0	0	0	1	1	1	1	1	1
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

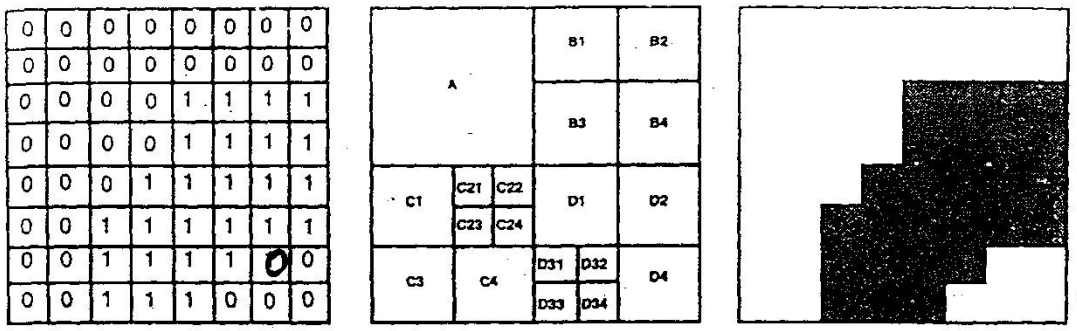
Quadtree Example

1	1	0	1	1	1	0	0	0	1	1	1	1	1	1	1
1	1	0	1	1	1	0	0	0	1	1	1	1	1	1	1
1	1	0	0	0	0	0	0	0	0	1	1	1	1	1	1
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Quadtree Example

1	1	0	1	1	1	0	0	0	1	1	1	1	1	1	1
1	1	0	1	1	1	0	0	0	1	1	1	1	1	1	1
1	1	0	0	0	0	0	0	0	0	1	1	1	1	1	1
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Quadtree Representation



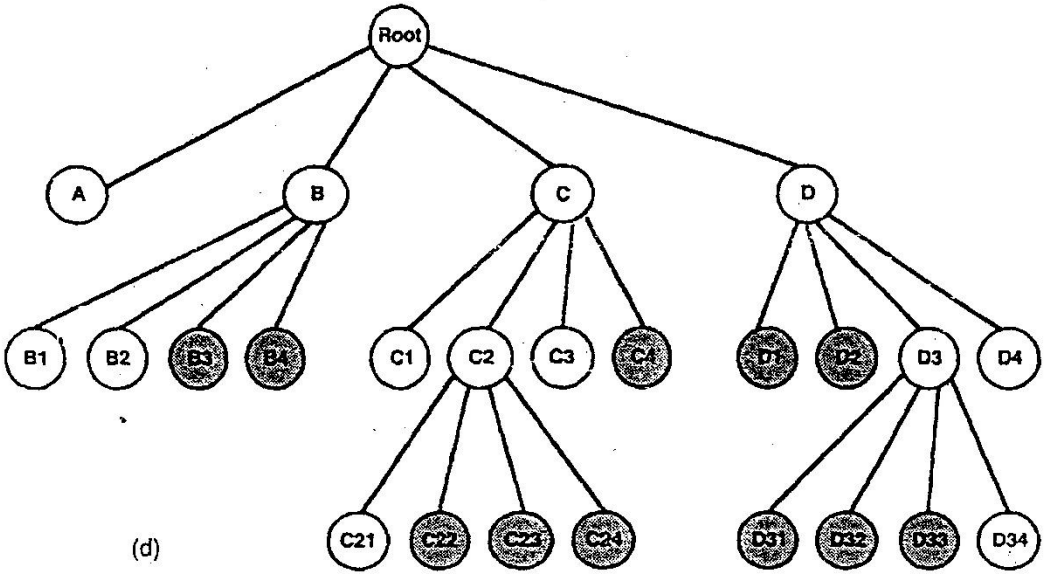
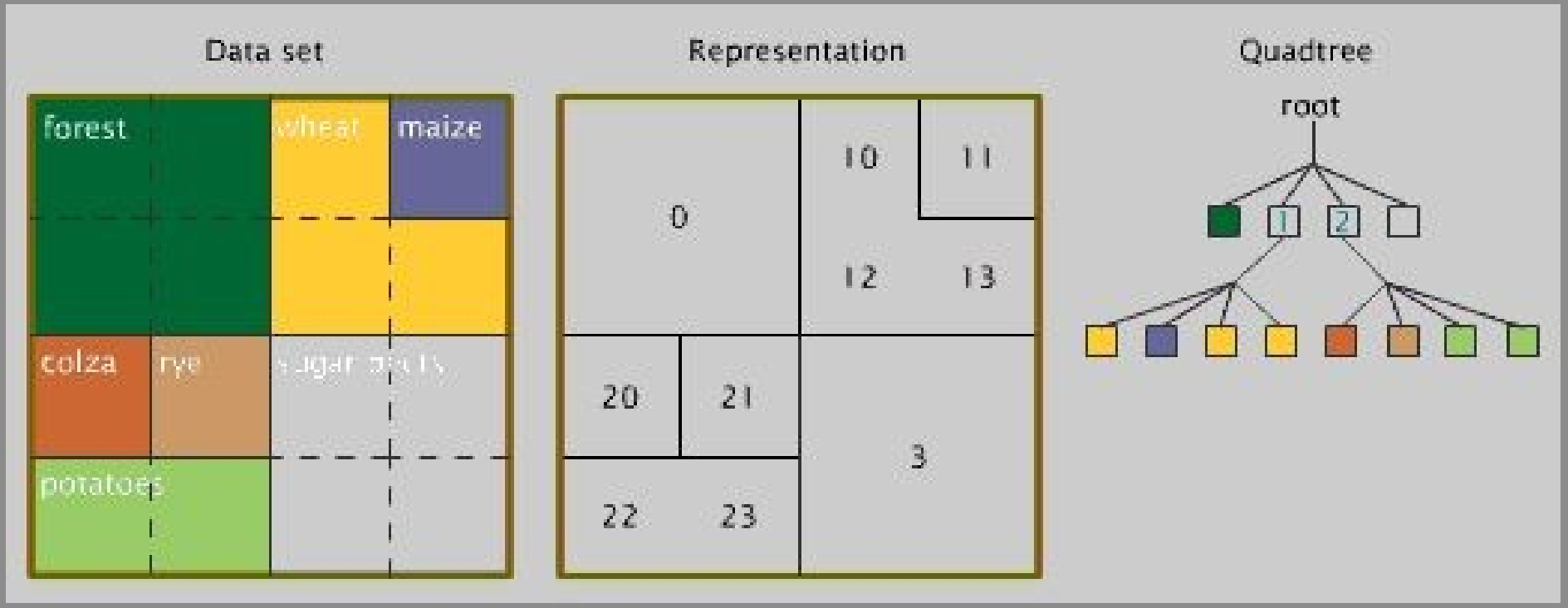


Figure 5.10 The quad-tree representation of a binary image

Quadtree Representation



Quadtree Representation

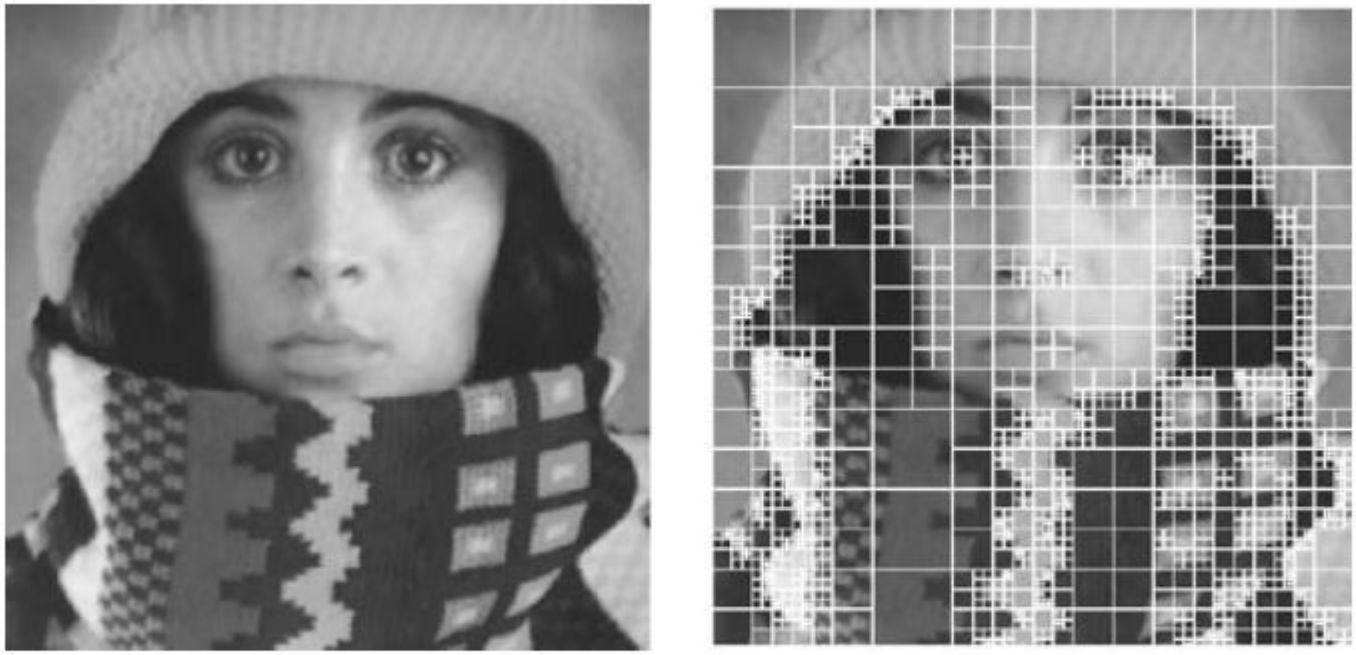
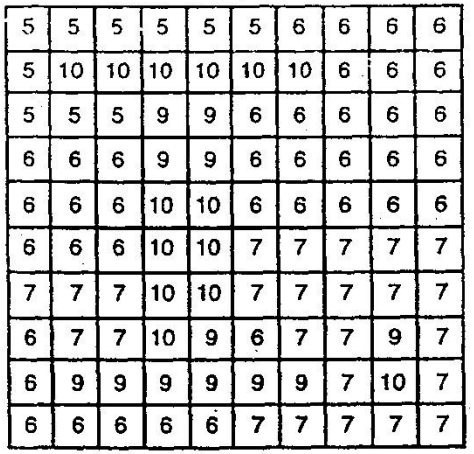
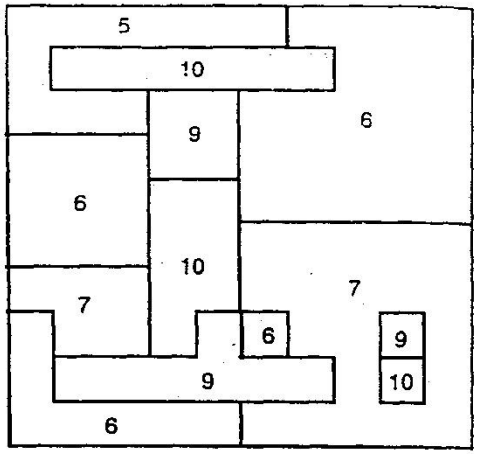


Figure 10.3 The image on the right delineates the regions obtained by carrying out a quadtree decomposition on the image using a similarity criterion based on the range of the intensity values within the blocks

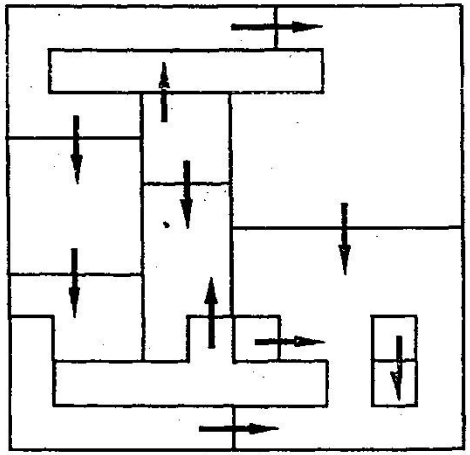
Region Merging & Splitting



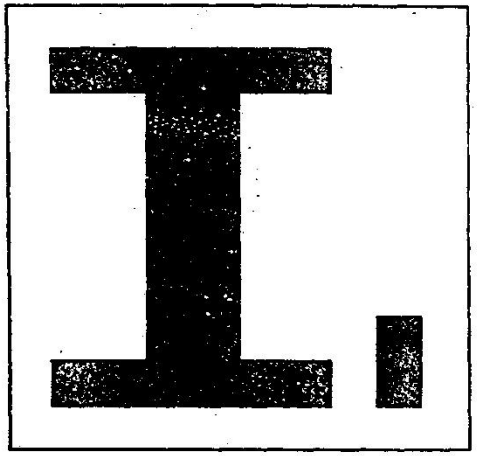
(a)



(b)



(c)



(d)

Figure 5.9 Region merging

The Split-and-Merge Algorithm

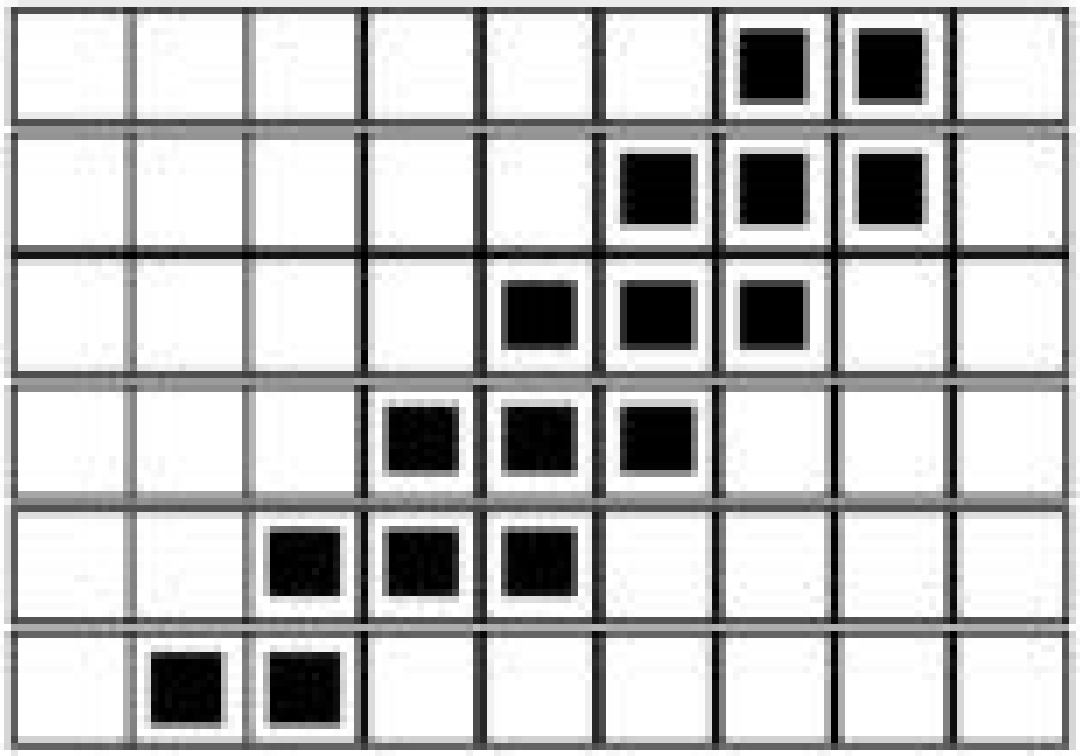
1	1	1	1	1	1	1	2
1	1	1	1	1	1	1	0
3	1	4	9	9	8	1	0
1	1	8	8	8	4	1	0
1	1	6	6	6	3	1	0
1	1	5	6	6	3	1	0
1	1	5	6	6	2	1	0
1	1	1	1	1	1	0	0

Sample image

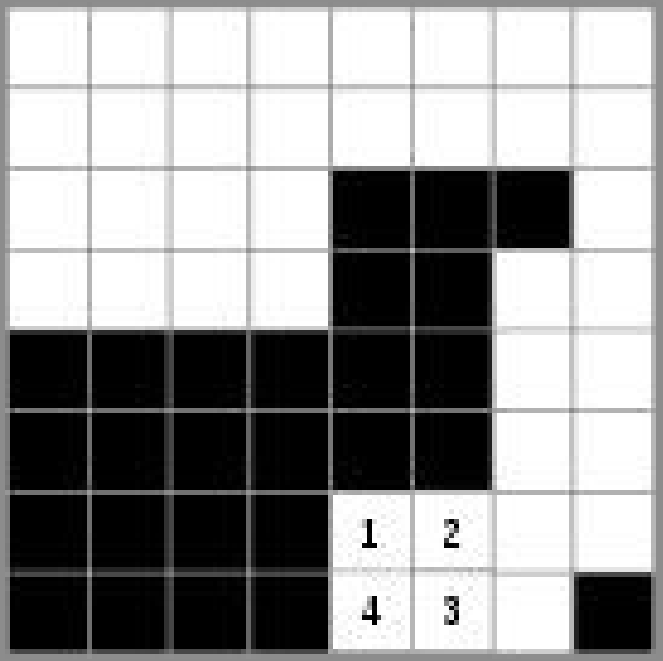
1	1	1	1	1	1	1	2
1	1	1	1	1	1	1	0
3	1	4	9	9	8	1	0
1	1	8	8	8	4	1	0
1	1	6	6	6	3	1	0
1	1	5	6	6	3	1	0
1	1	5	6	6	2	1	0
1	1	1	1	1	1	0	0

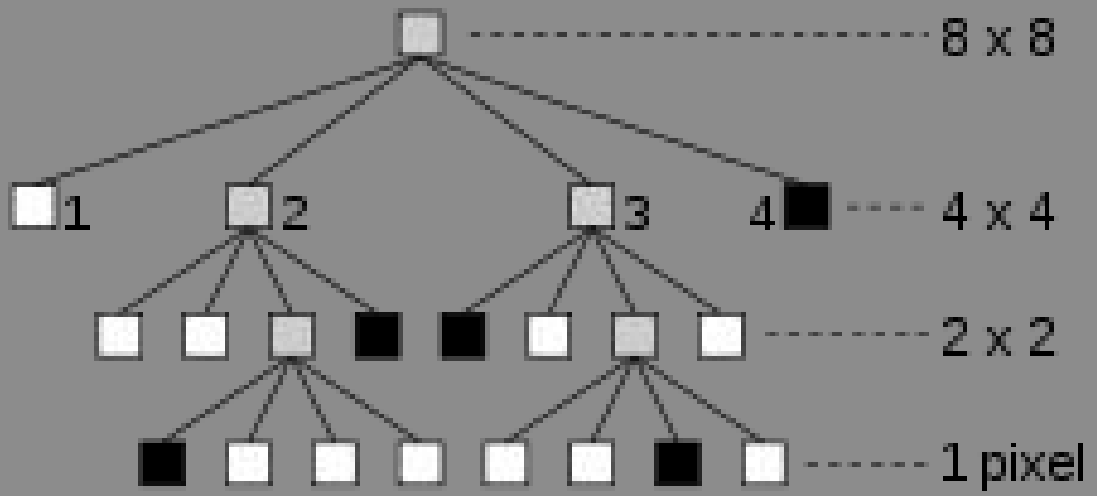
First split

RLE

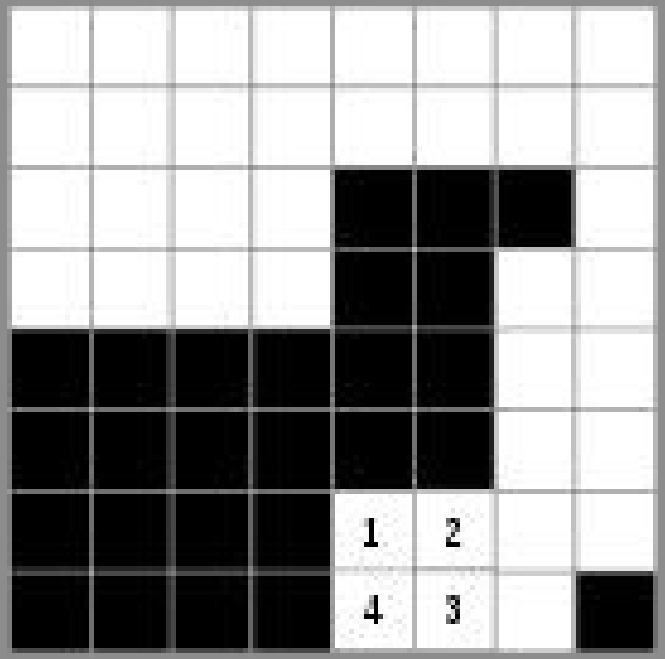


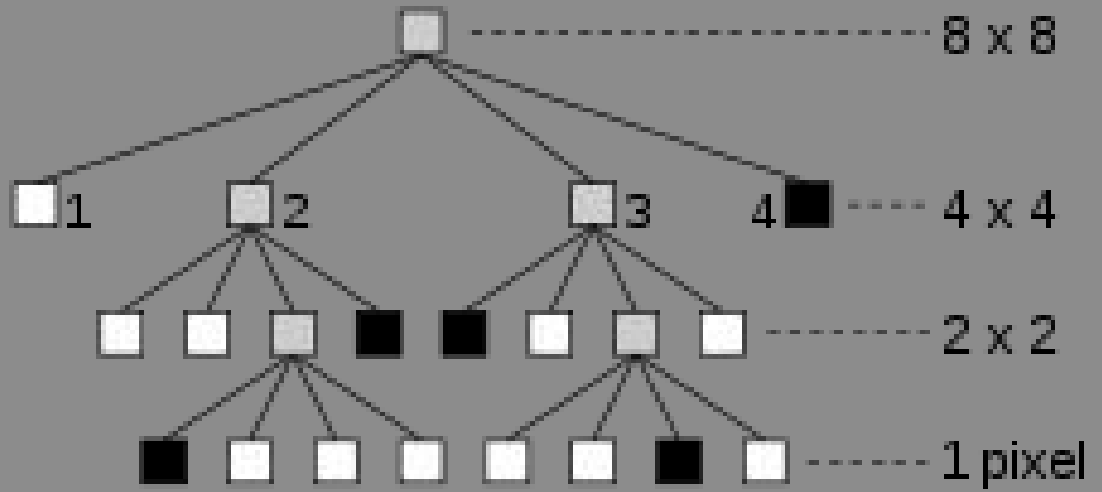
Quadtree Decomposition



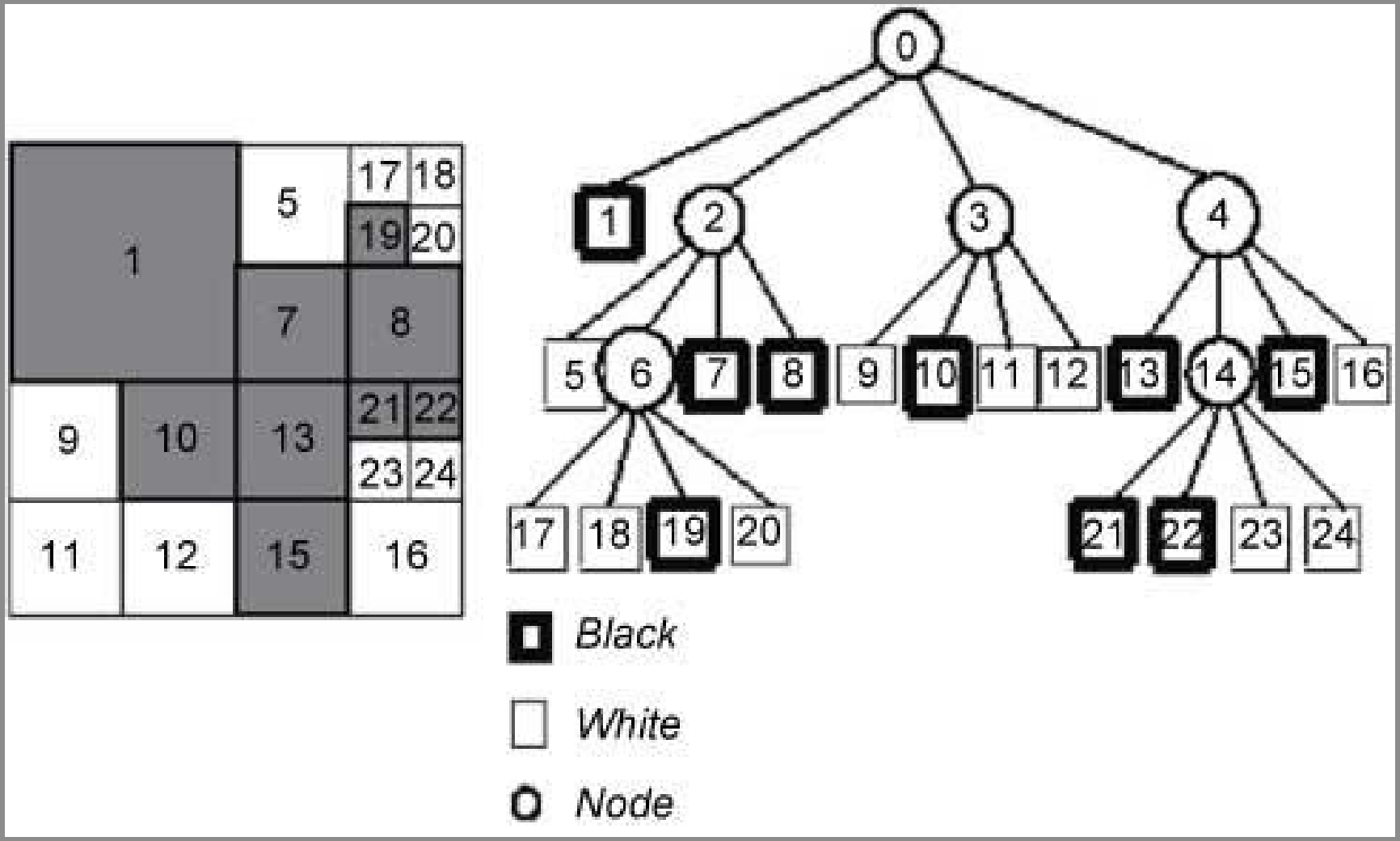


Quadtree Decomposition

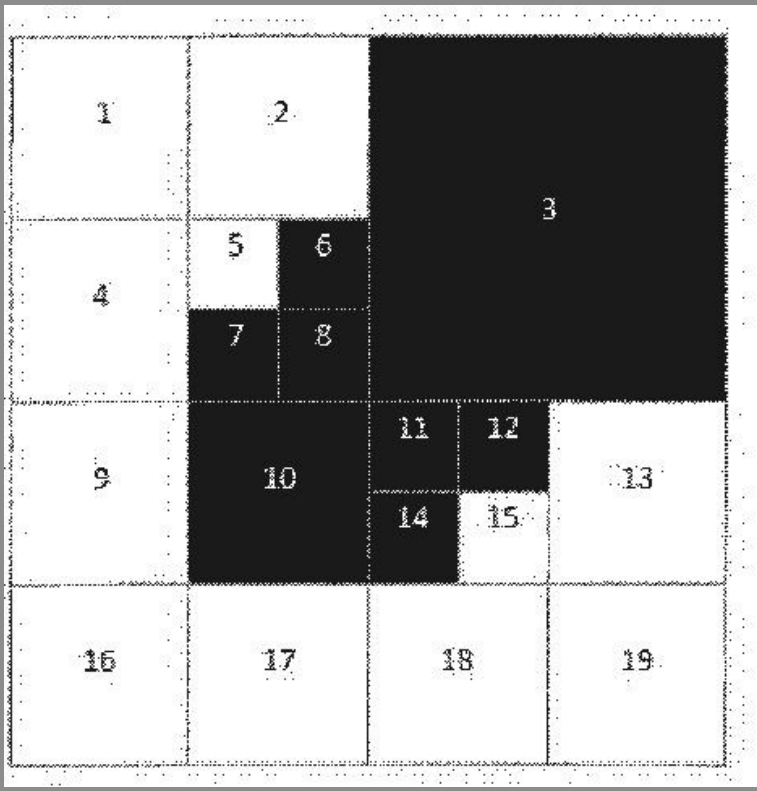


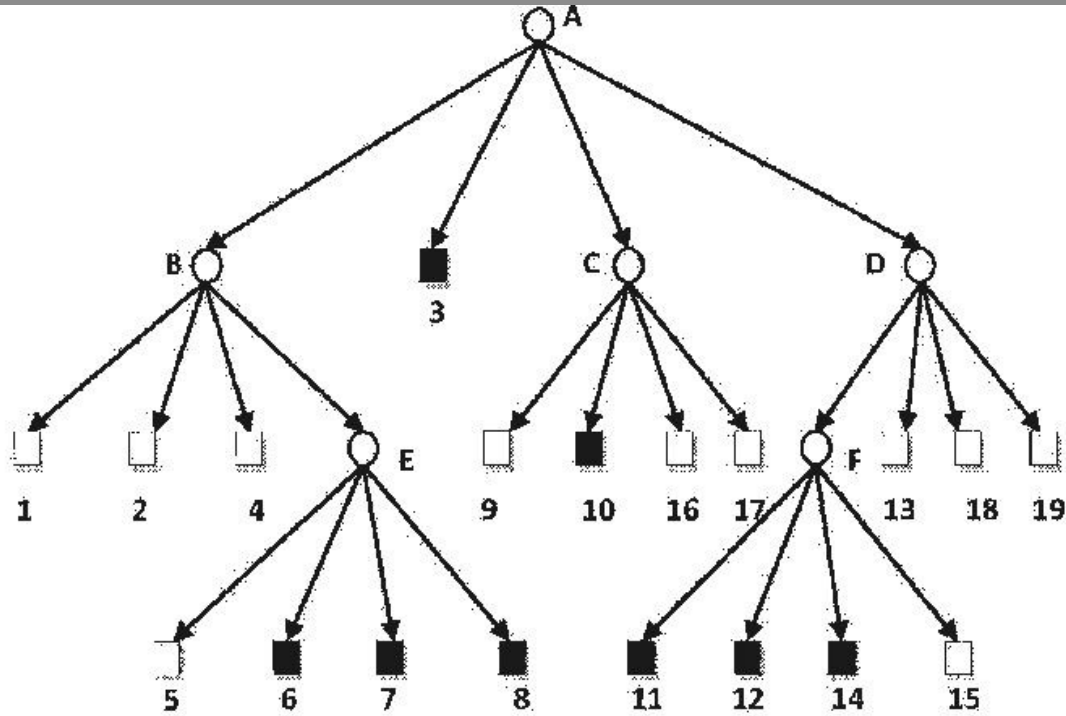


Quadtree Decomposition

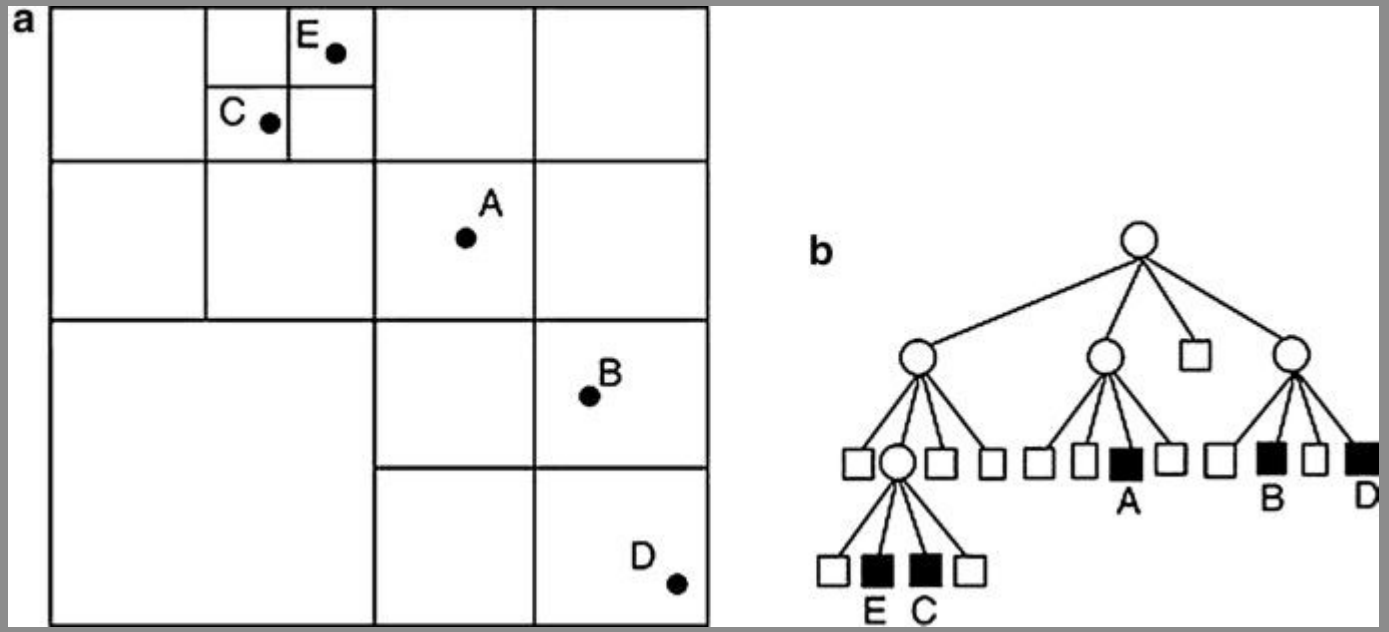


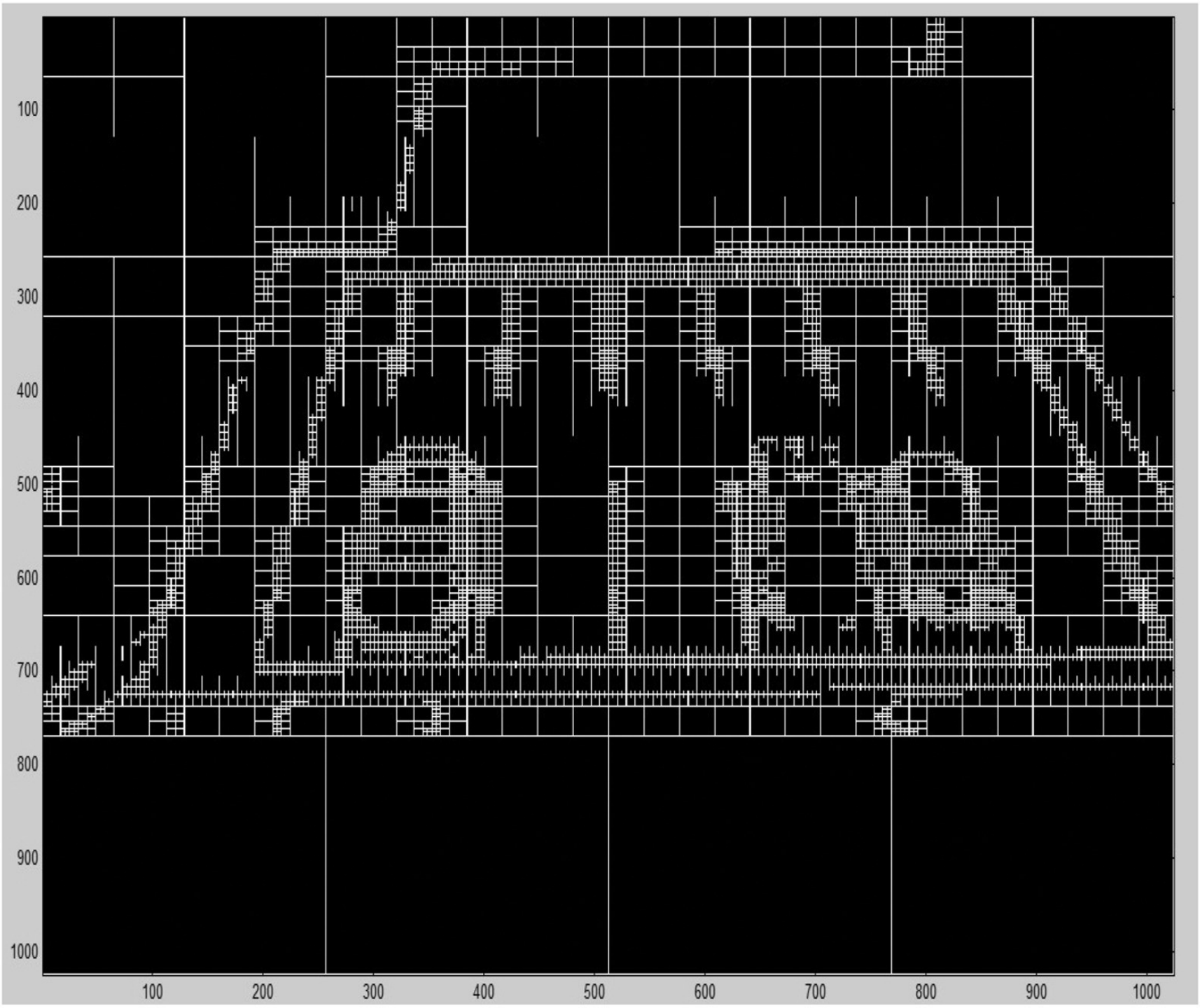
Quadtree Decomposition





Quadtree Decomposition




100
200
300
400
500
600
700
800
900
1000
100
200
300
400
500
600
700
800
900
1000